# Design, Development and Realization of Quadratic Volterra Filters for Selected Applications

Thesis submitted in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy in Engineering under the Faculty of Engineering

By

Hari V S
(Reg. No. 3759)

Under the guidance of

**Prof. (Dr.) Jagathy Raj V P**
(Supervising Guide)

and

**Prof. (Dr.) Gopikakumari R**
(Co-Guide)


**School of Engineering**
**Cochin University of Science and Technology**
**Kochi - 682 022**

**June 2013**

# Design, Development and Realization of Quadratic Volterra Filters for Selected Applications

PhD Thesis under the Faculty of Engineering

**Author**
Hari V S
Research Scholar
School of Engineering
Cochin University of Science and Technology
E-mail - hari_cec@yahoo.com

**Supervising Guide**
Prof. (Dr.) Jagathy Raj V P
School of Engineering
Cochin University of Science and Technology
E-mail - jagathy@cusat.ac.in

**Co-Guide**
Prof. (Dr.) Gopikakumari R
School of Engineering
Cochin University of Science and Technology
E-mail - gopika@cusat.ac.in


**School of Engineering**
**Cochin University of Science and Technology**
**Kochi - 682 022**


School of Engineering
Cochin University of Science and Technology
Kochi - 682 022

# Certificate

Certified that the thesis titled *Design, Development and Realization of Quadratic Volterra Filters for Selected Applications* is a *bona fide* record of the research work carried out by Mr. Hari V S under our supervision in Faculty of Engineering, Cochin University of Science and Technology. The work presented in this thesis or part thereof has not been presented for any other degree.

Prof. (Dr.) Jagathy Raj V P
(Supervising guide)

Prof. (Dr.) Gopikakumari R
(Co-guide)

# Acknowledgment

Research that leads to PhD is once in a lifetime experience and it does not come to fruition without the help and support from many, which need be acknowledged. Let me acknowledge the help rendered by the Director, Institute of Human Resources Development, for permitting me to do research along with my official responsibilities. Thanks are due to the Vice Chancellor, Controller of Examination and other officials of Cochin University of Science and Technology for selecting me to carry out my research.

It was undoubtedly the guidance and advice of my research guide Prof. Jagathy Raj V P that carried me along. His helpful interventions were not just limited to the accomplishment of work but extended into proper presentation of work, preparation of papers and the choice of proper methodologies. His zeal and devotion to academics is a source of inspiration in my teaching career as well.

The help and support from my co-guide Prof. Gopikakumari R in conducting the work, especially in image processing, had no parallel. Despite her busy schedule, she found time to interact in the work, to point out flaws and to correct the papers, presentations and reports. Let me express my heartfelt gratitude to her at this opportunity.

Let me extend sincere thanks to the Principal, School of Engineering for providing the proper resources for research. I am grateful to the

office staff of School of Engineering and Cochin University for accomplishing the background paper work without delay.

Let me express my sincere gratitude to my colleagues at Institute of Human Resources Development, friends and students for their timely help and suggestions. Heartfelt thanks are due to my family, especially my wife Nisha and my daughter Gayathri, who helped me in this venture by sparing the many daily hassles that otherwise would have obstructed my prolonged reflections and work.

Hari V S

# Preface

The basic concepts of *digital signal processing* are taught to the students in engineering and science. The focus of the course is on linear, time invariant systems. The question as to what happens when the system is governed by a quadratic or cubic equation remains unanswered in the vast majority of literature on signal processing. Light has been shed on this problem when John V Mathews and Giovanni L Sicuranza published the book *Polynomial Signal Processing*. This book opened up an unseen vista of polynomial systems for signal and image processing. The book presented the theory and implementations of both adaptive and non-adaptive FIR and IIR quadratic systems which offer improved performance than conventional linear systems.

The theory of quadratic systems presents a pristine and virgin area of research that offers computationally intensive work. Once the area of research is selected, the next issue is the choice of the software tool to carry out the work. Conventional languages like C and C++ are easily eliminated as they are not interpreted and lack good quality plotting libraries. MATLAB is proved to be very slow and so do SCILAB and Octave. The search for a language for scientific computing that was as fast as C, but with a good quality plotting library, ended up in Python, a distant relative of LISP. It proved to be ideal for scientific computing. An account of the use of Python, its scientific computing package *scipy* and the plotting library *pylab* is given in the appendix.

Initially, work is focused on designing predictors that exploit the polynomial nonlinearities inherent in speech generation mechanisms. Soon, the work got diverted into medical image processing which offered more potential to exploit by the use of quadratic methods. The major focus in this area is on quadratic edge detection methods for retinal images and fingerprints as well as de-noising raw MRI signals.

The organization of the work is as outlined in Sec. 1.4 in page 20. The reader is advised to read this section before proceeding with the remaining chapters. The first two chapters give a concise introduction to quadratic systems. Chapter 3 gives a detailed account of the methodology of research adopted. The general design and implementation strategies in the research work are detailed in chapter 4. The chapters from 5 to 8 detail the research work done in various applications. These chapters invariably contain a section devoted to the methodology of research for that specific application. Chapter 9 summarizes the work, highlighting the key merits of quadratic systems. The research contributions and the impacts to various stakeholders are discussed in Chapter 10. Every effort is made to enhance the readability of the thesis by adding as many post-script pictures as possible. The structure of the document is made easily navigable by including ample references, cross-references and mini-table of contents in every chapter. With this, it is hoped that the reader can go through the chapters and understand the research work on quadratic systems.

*Hari V S*

# Abstract

Most of the present day signal processing strategies revolve around linear time invariant systems. Though these methods are popular and useful in many applications, their performance degrades when the effects of nonlinear components are dominant. Polynomial methods are sought as an alternative in this research work. The work aims at modeling polynomial nonlinear phenomena present in real time systems and signals employed for speech and image processing. Volterra power series that realizes polynomial filters as combinations of quadratic, cubic and higher order systems with linear systems is studied. The work focuses on quadratic systems that can model the nonlinear effects in the generation and processing of speech and image signals, with the objective of achieving better performance parameters than those offered by linear systems and other nonlinear systems. Special design methodologies are employed for designing quadratic filters. The computational challenges arising from the usage of quadratic systems are overcome by special hardware implementation schemes.

Multiple reflections, in the vocal cavity during speech generation, add polynomial products in the signal which cannot be modeled by conventional linear predictors. Statistical prediction of such signals are identified as a problem where the potentials of quadratic filters are exploited. Quadratic predictor, based on optimization method, is designed and realized. The predictor is put to use in the differential speech coder and decoder. The modified DPCM system yields re-

duced mean square error between the transmitted and received signal.

Images are captured by nonlinear systems and the human visual system that perceives the image is also inherently nonlinear and complex. Hence processing images is a fertile arena for nonlinear signal processing. One of the major areas of image processing and computer vision is the detection of edges or peripheries. Edges are caused by sharp discontinuities in pixel values that are detected by several types of nonlinear filters. Often, these methods are not precise and are sensitive to noise. To overcome these demerits, quadratic filters are designed and implemented for better edge detection in presence of noise. Two dimensional version of the Teager algorithm is used to implement type-II filters for edge detection. The quadratic filters designed are put to use in the enhancement of fingerprints and the detection of retinal microaneurysms due to diabetic retinopathy.

Linear system theory offers well established methods to remove additive noise from the desired signal, especially when the noise has Gaussian statistics. The noise that corrupts natural images is rarely Gaussian but impulsive in nature. Quadratic filter is designed and implemented for removing impulsive noise from raw MRI data. This filter offers substantial improvement in the signal to noise ratio, when compared with other nonlinear filters like median and minimum filters.

Comparisons of performance parameters with conventional systems establish the superiority of quadratic systems for the selected applications. These applications have impact on both researchers and practitioners in the fields of fingerprint recognition, ophthalmology, oncology and electronic communication and forensic sciences.

# Contents

# List of Figures

# List of Tables

# Part I

# Overture

# Foreword to Part I

Discrete signal processing impacts many walks of present day life. Though many ramifications of signal processing are studied and worked on, polynomial signal processing is a relatively new avenue, which the research work ventures into. Part I contains four chapters. Chapter 1 gives a brief overview of signal processing and lists the key objectives of the research. It also outlines the general methodology of research adopted and the organization of the thesis.

Chapter 2 involves the survey of existing literature pertaining to signal processing. It stresses on the necessity of polynomial signal processing and discusses the various power series expansions, especially the Volterra series. Both 1-D and 2-D quadratic systems and their key properties are discussed in this chapter.

The scheme of work and methodology are presented in chapter 3. It details the research problems and discusses the strategies to approach each problem. The chapter presents the scope of work too.

The strategies for design and computationally efficient implementations of quadratic filters are presented in chapter 4. Comparisons between various design and implementation methods are performed to select specific methodologies suitable for the work.

Chapter 1

# Introduction

## Contents

## 1.1 Introduction

Digital signal processing is a science that involves the manipulation of sequences of numbers or symbols to modify or improve them in some way. Discrete signal processing(DSP) has played a major role in the developments in communication systems, biomedical data processing, digital audio etc.[Oppenheim and Schafer 1998],[Roberts and Mullis 1987]. The biggest breakthrough in signal processing was the introduction of efficient algorithms for the computation of Fourier transform[Singleton 1967], [Singleton 1969], [Winograd 1976]. Many methods were proposed[Bergland 1968],[Bluestein 1970] of which the reinvention of Gauss's method by Cooley and Tuckey became widely accepted[Cooley 1965],[Heideman and Burrus 1984]. Traditionally, discrete linear systems have been extensively used for processing both one dimensional and two dimensional digital signals. These linear filtering models are validated by the assumption of statistical stationarity and Gaussianity of the signals processed[Athanasios Papoulis 2002],[Hayes 2003],[Mix 1995]. Such systems process the incoming signal as they are, without consideration of the signal generation mechanisms, on the assumption of linearity and time invariance but render ease of design and easy implementation on DSP hardware or FPGA[Mayer-Baese 2007].

Discrete time systems are broadly classified into two categories.

- Finite impulse response (FIR) filters and
- Infinite impulse response (IIR) filters

Finite impulse response LTI systems, whose impulse responses are finite in duration, have been extensively used in conventional filtering, statistical estimation and prediction. Infinite impulse response filters are discretized versions of conventional analog filters and are popular

in frequency selective filtering. Though these linear systems can perform much of signal processing tasks, they fail where signals are neither stationary nor Gaussian as in the case of images. Besides, when one desires to take into account the effects of nonlinearities present during signal generation or within systems, one has to look beyond linear time invariant systems and reach into nonlinear signal processing.

Nonlinear filters such as order statistic filters, homomorphic filters, fuzzy filters etc. are given preference in two dimensional signal processing with a view that human visual system is nonlinear. Although they perform better than the linear counterparts in image processing tasks, most of these filters are very sensitive to noise. The greater part of the nonlinear effects are confined to the quadratic polynomial content, which are not modeled by the conventional nonlinear filters. So it becomes necessary to consider systems that have polynomial relationship between input and output, based on a suitable power series, that can outperform linear systems and other conventional nonlinear systems. These objectives are detailed in Sec. 1.2.

## 1.2 Objectives

The main objective of the research work is to design and implement quadratic filters, based on Volterra series, for practical applications. More specifically, the objectives of work are elucidated as below.

- To study and compare various linear and nonlinear systems.
- To study various polynomial power series expansions and to do a suitability analysis.
- To develop a general design strategy for the development of quadratic Volterra filters.

- To formulate the implementation strategy for quadratic filters.
- To develop a conceptual structure for the practical realization of quadratic filters.
- To accomplish the specific design and implementation of quadratic filters for practical applications.
- To make a comparison study of quadratic Volterra filters with conventional linear and nonlinear filters in terms of various performance parameters for the above applications.

## 1.3 Methodology

The research work that is to be carried out for accomplishing the objectives in Sec. 1.2 follows the methodology as depicted in Fig. 1.1. First phase of the work is the study of linear systems and identification of its limitations when effects of nonlinearity are present.

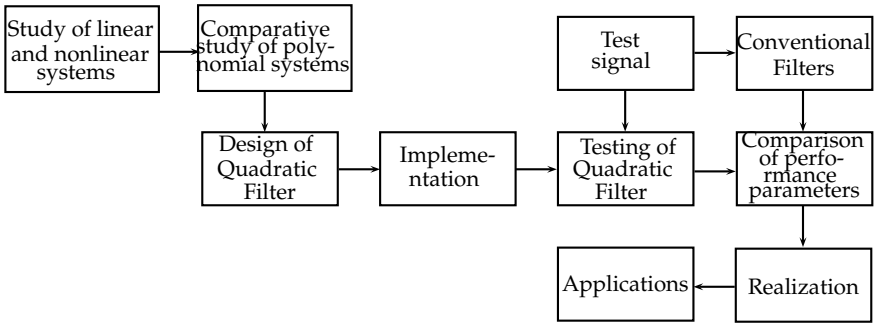



**Fig. 1.1.** Methodology

Second phase of the work involves in contrasting polynomial systems with linear systems and other nonlinear systems in modeling mild polynomial nonlinearities. During this phase, quadratic systems are

reached at as a potential choice for modeling mild polynomial nonlinearities. The third phase of work is the design, whose strategies are presented in Sec. 4.2 in page 51, of quadratic filter for the specific application. The structure of quadratic filters being more complex, careful approximations need be done for practical implementations which are discussed in Sec. 4.3 in page 53. This forms the fourth phase of work in every practical application. The outcomes of research are validated by testing with known test signals and assessing the performance parameters. In the fifth phase of work, the same test signals are used to excite conventional filters intended for the same application and the same performance parameters are computed for comparison with quadratic filters. Computationally efficient practical realizations of specific designs are carried out in the sixth phase of work. In the last phase of work, the validated quadratic filter is put to use in the specific application. The detailed methodology is explained in chapter 4.

## 1.4 Outline of Thesis

The graphical abstract in Fig. 1.2 shows the structure of the thesis and the contents in each chapter. The encircled numbers in this diagram indicate the chapter numbers. The thesis is divided into five parts viz.

1. Part I - Overture
2. Part II - Quadratic Volterra Filters for edge detection
3. Part III - Quadratic Volterra Filters for noise removal
4. Part IV - Quadratic Volterra Filters for nonlinear prediction
5. Part V - Summary and Research Contributions

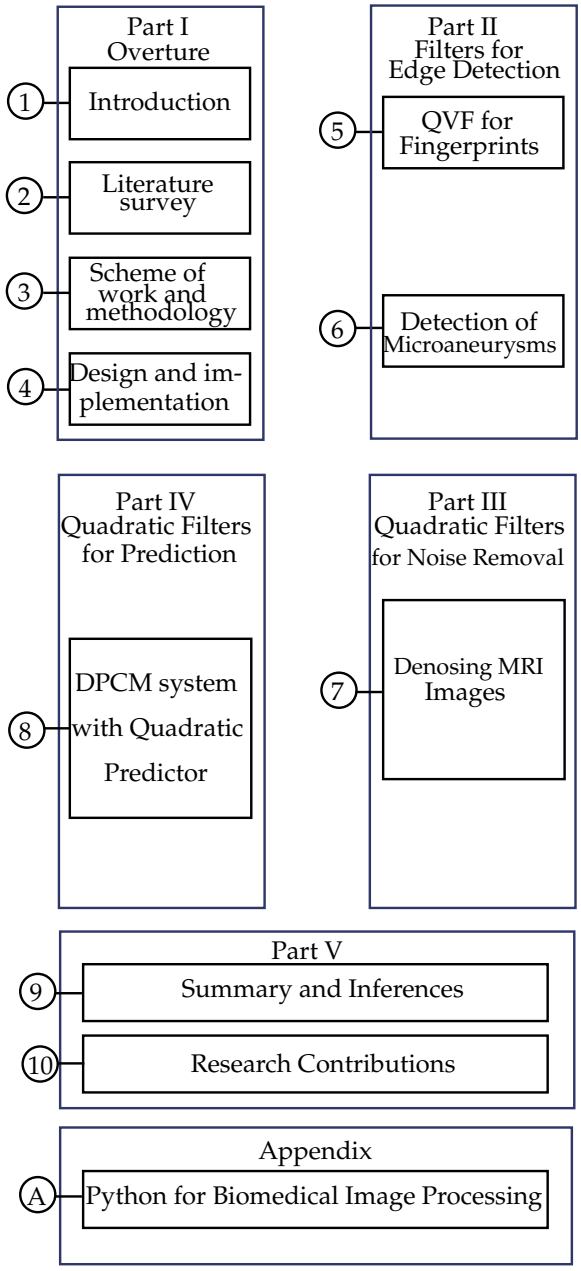



**Fig. 1.2.** Graphical abstract of thesis

### 1.4.1 Part I - Overture

Chapter 1 in this part gives a general introduction to linear signal processing and its limitations in modeling nonlinear systems. It elucidates the objectives of the research work. The concept of Volterra series, quadratic Volterra systems and their properties are presented in chapter 2. The scheme of work and methodology of research adopted are detailed in chapter 3. The design methodologies and implementation strategies for quadratic Volterra filters are presented in chapter 4.

### 1.4.2 Part II - Quadratic Volterra Filters for Edge Detection

The second part of the thesis contains two chapters that discuss the design and implementation of the following quadratic filters.

- QVF for enhancing noisy fingerprints is presented in chapter 5
- QVF for detecting microaneurysms due to diabetic retinopathy is presented in chapter 6.

### 1.4.3 Part III - Quadratic Volterra Filters for Noise Removal

MRI relies on the Fourier transformation of nuclear magnetic resonances for the formation of the image of the body part under study. The acquisition of MRI data happens under strong magnetic field and consequently the data is corrupted by impulsive noise. Chapter 7 in this part discusses the design, implementation and testing of quadratic filter for the removal of impulsive noise from raw MRI data.

### 1.4.4 Part IV - Quadratic Volterra Filters for Statistical Prediction

Chapter 8 in the fourth part of the thesis deals with the design and development of quadratic predictors for speech. Two quadratic methods in addition to the linear method of speech prediction are presented. Differential pulse code modulation systems employing the three predictors and the comparison of their performance are also discussed.

### 1.4.5 Part V - Research Contributions, Summary and Conclusion

Chapter 9 begins with a review of the work and concludes the work done for practical applications with stress on comparison of figures of merit on employing quadratic filters with linear and other nonlinear filters. The limitations of the present work and the possible future expansions are also presented in this chapter. Chapter 10 opens with an overview of the impact of the applications of digital signal processing and discusses the objectives and motivation for research in quadratic systems. The chapter details the impact of the research on various stakeholders.

### 1.4.6 Appendix A: Python - A Computational Tool for Biomedical Image Processing

The appendix details the use of the object oriented, interpreted language Python and its scientific computing toolbox, Scipy, with which much of the computations in this research work are accomplished. It stresses on Python modules for image processing for bio-medical applications and is included for quick reference, taking into consideration the scarcity of learning material on Python programming and the lack of awareness of its suitability in scientific computing.

Chapter 2

# Literature Survey

## Contents



## 2.1 Overview

Discrete signals and systems play vital roles in today's engineering domains such as instrumentation, communication, control systems, medical imaging, radar systems etc. and maintain a steady growth

in cost effective usage and applications. However, most of the theory and research work are focused around linear and time invariant (LTI) systems, owing to the ease of design and simplicity of implementation. But when signals generated by or systems governed by mild polynomial nonlinearities are encountered, one need to consider the usage of polynomial systems which are modeled as extension of the conventional LTI systems. Obviously, one has to resort to mathematical models that link the input and output in the form of a power series that can add quadratic, cubic and higher order components in parallel with the linear term. Although Taylor series is used for such a model, it suffers from lack of memory and huge computational complexity. Sec. 2.2 outlines the LTI systems and their limitations in modeling nonlinearities and Sec. 2.3 discusses Taylor series and its shortcomings. A variant of Taylor series which possesses memory is Volterra series, which is employed to model polynomial relationship between system input and output, the details of which are presented in Sec. 2.4. Most of the nonlinear effects are encompassed by the quadratic term and hence quadratic systems, both one dimensional and two dimensional, are presented in Sec. 2.4, 2.5 and 2.6. The design methodologies for quadratic systems are presented in Sec. 4.2. The computational complexity arising from the use of Volterra series is surmounted by various strategies for realization as presented in detail in Sec. 4.3.

## 2.2 Linear Time Invariant Systems

Linear time invariant systems are those that obey superposition and shift invariance properties. The output $y[n]$ of a linear and time invariant(LTI) system [Oppenheim and Schafer 1998] driven by input $x[n]$ is given as

$$y[n] = \sum_{k=-\infty}^{\infty} h[k]x[n-k] \tag{2.1}$$

Eq. 2.1 is denoted as $y[n] = x[n] * h[n]$ where the $*$ sign indicates the linear convolution operation. The parameter $h[n]$ is the kernel or the impulse response of the LTI system that characterizes the system completely. The essential problem in the design of a discrete filter as in Eq. 2.1 is in ascertaining the kernel $h[n]$ for a given input $x[n]$ and for a desired output $y[n]$. This problem is relatively mild when the duration of $h[n]$ is infinite. In this case, discretization of conventional analog filters such as Chebyshev, Bessel filters etc. are possible, resulting in frequency selective infinite impulse response (IIR) filters. When the duration of the kernel $h[n]$ is finite, the design is more involved. Methods based on statistical correlations are proposed[Hopf 1934],[Wiener 1956],[Noble 1988]. Optimization strategies are also proposed for FIR filter design[McClellan and Parks 2005], [Adams 1991], [Adams et al. 1993],[Kaiser 1983], [Hermann 1970], [Gold 1975]. Frequency sampling methods for finding $h[n]$ are available in literature [Rabiner and Parks 1967], [Rader and Gold 1975].

When it comes to two dimensional signal processing, much of the work is focused around natural images. A natural image is a two dimensional signal rendered as an $M \times N$ array of real numbers by a digital camera [Ekstrom 1984],[Gonzalez and Woods 1992]. The generation of the image is a nonlinear process and so is the human visual system that perceives the image. Processing the image involves manipulation of the two dimensional array of real numbers from the camera, say $x[n_1, n_2]$, to yield a more useful array of numbers, say $y[n_1, n_2]$, by a two dimensional filter kernel $h[n_1, n_2]$ as given in Eq. 2.2.

$$y[n_1, n_2] = \sum_{k_1=0}^{N-1} \sum_{k_2=0}^{M-1} h[k_1, k_2] x[n_1 - k_1, n_2 - k_2] \tag{2.2}$$

Eq. 2.2, denoted as $y[n_1, n_2] = x[n_1, n_2] * *h[n_1, n_2]$, is a two dimensional linear convolution of the input with the linear filter kernel. This

step finds extensive application in space exploration, navigation, medical diagnosis, multimedia communication etc. The tasks in image processing are segmentation, edge detection, classification, noise removal etc. Extraction of features is a key issue, especially in medical image processing [Jan 2005], [Dougherty 2011]. As the image generation as well as the image perception is nonlinear, it is logical to employ nonlinear filters for processing images. The major nonlinear systems, categorized as in Fig. 2.1, are

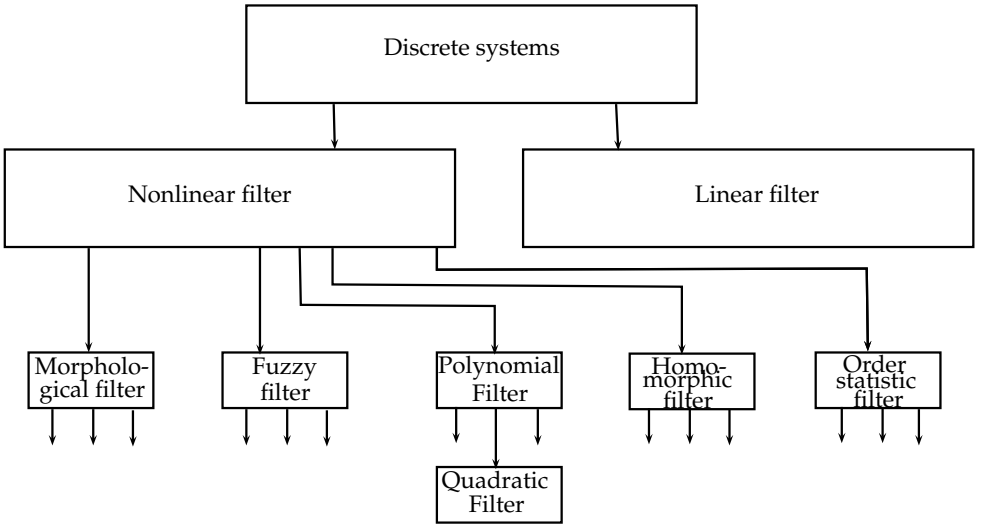



**Fig. 2.1.** Broad classification of nonlinear systems

- Homomorphic filters
- Order statistic filters
- Morphological filters
- Fuzzy filters

- Polynomial filters

**Homomorphic Filters** are useful when the input signal is a multiplicative combination of two different signals. The two components are processed in different ways using a nonlinear transformation, followed by linear filtering. The filter is popularly used in isolating multiplicative noise and in improving the contrast with normalized brightness.

**Order Statistic Filters** are used for removing impulsive noise from the input image without blurring its edges. Median, weighted median and mean filters are popular members in this category.

**Morphological Filters** employ geometric or morphological transformations on input image to extract the skeletal sketches of objects in the image. These morphological operations rely only on the relative ordering of pixel values and not on their numerical values, and therefore are especially suited to the processing of binary images.

**Fuzzy Filters** apply fuzzy reasoning to model the uncertainty in the image and are employed in filtering, interpolation and morphology.

While all the above methods are used in various image processing applications, polynomial systems have the distinctive advantage that they can model a larger class of nonlinear systems with a smaller number of coefficients. Besides they are constructed as extensions of a linear system by suitable power series expansion. The popular Taylor series is deselected based on reasons that will be elaborated in Sec. 2.3. In its stead, Volterra power series that can add quadratic, cubic and higher order systems in parallel with the linear system is selected. The work focuses on one dimensional and two dimensional quadratic systems based on Volterra series for practical applications.

The details of Volterra systems and the work done for realization for various applications are detailed from chapters 2 to 8 as outlined in the next section. Now that the idea of systems following a polynomial relationship between input and output is conceived, it is imperative to look for a general mathematical framework for polynomial systems. It is intuitive to consider Taylor series [Struik 1969] for this purpose.

## 2.3 Taylor Series

The power series is proposed by Brook Taylor in 1715 for expressing the function $y$ as

$$y = f(x) = \sum_{i=0}^{\infty} \frac{f^{(n)}(a)}{i\,!}(x-a)^i \tag{2.3}$$

where $f^{(n)}(a)$ denotes the $n^{\text{th}}$ derivative of $f$ at the point $a$. When $a = 0$, the series is called Maclaurin series. Even though Taylor series provides a polynomial relationship between $x$ and $y$, it cannot easily be employed for modeling polynomial systems due to the following shortcomings.

- Taylor series, being a memoryless series,[Sicuranza 1992] only the present value of the input $x$ but not its history is incorporated in the model.
- The convergence being slow, more coefficients and powers are needed for the approximation of $y$, resulting in huge complexity.
- Practical implementations of polynomial systems are much more convenient if square, cubic and higher powers are added with Eq. 2.1 as this enables the designer to add quadratic, cubic or higher order systems in parallel with the existing LTI system. Such a representation is not possible with Taylor series.

The above points warrant another series representation for polynomial systems that can augment Eq. 2.1 with quadratic, cubic and higher order systems. Such a power series was proposed by Vito Volterra, the discrete version of which is discussed in the next section.

## 2.4 Discrete Volterra Series

Vito Volterra proposed a power series for describing the input - output relation of an $N^{\text{th}}$ order nonlinear system[Alper 1963], the discrete version of which is given as

$$y[n] = h_0 + \sum_{r=1}^{\infty} \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} \cdots \sum_{n_r=1}^{N} h_r[n_1, n_2, \ldots n_r]. \qquad (2.4)$$
$$x[n-n_1]x[n-n_2]\cdots x[n-n_r]$$

The term $h_0$ denotes the output offset when no input is present and the term $h_r[n_1, n_2, \ldots n_r]$ denotes the $r^{\text{th}}$ order Volterra kernel. Identification of this kernel for a nonlinear system is the chief issue in designing polynomial systems. By employing Volterra series, one can surmount many disadvantages with Taylor series for modeling polynomial systems. With order $r = 1$, Eq. 2.4 defaults to a convolution between input and output, representing a linear system. For $r = 2$, it becomes a quadratic system given as [Mathews and Sicuranza 2000]

$$y[n] = h_0 + \sum_{m_1=1}^{N} h_1[m_1]x[n-m_1] + \sum_{n_1=1}^{N} \sum_{n_2=1}^{N} h_2[n_1, n_2]x[n-n_1]x[n-n_2] \qquad (2.5)$$

It has been observed that majority of the effects due to polynomial nonlinearities are confined to the quadratic term and so the interest

in quadratic systems. Eq. 2.5 is expressed as a matrix equation

$$y[n] = h_0 + \mathsf{X}_1^T[n]\mathsf{H}_1 + \mathsf{X}_1^T[n]\mathsf{H}_2\mathsf{X}_1[n] \tag{2.6}$$

where

$$\mathsf{X}_1 = [\, x[n], x[n-1], \cdots, x[n-N+1] \,]^T \tag{2.7}$$

and

$$\mathsf{H}_1 = [\, h_1[0], h_1[1], \cdots, h_1[N-1] \,]^T \tag{2.8}$$

The term $\mathsf{H}_2$ is expressed as

$$\mathsf{H}_2 = \begin{bmatrix} h_2[0,0] & h_2[0,1] & \cdots & h_2[0,N-1] \\ h_2[1,0] & h_2[1,1] & \cdots & h_2[1,N-1] \\ h_2[2,0] & h_2[2,1] & \cdots & h_2[2,N-1] \\ \vdots & \vdots & \ddots & \vdots \\ h_2[N-1,0] & h_2[N-1,1] & \cdots & h_2[N-1,N-1] \end{bmatrix} \tag{2.9}$$

This system is realized as a parallel combination of three components as in Fig. 2.2. The quadratic system is defined by the third term in Eq. 2.6 is

$$y_q[n] = \mathsf{X}_1^T[n]\mathsf{H}_2\mathsf{X}_1[n] \tag{2.10}$$

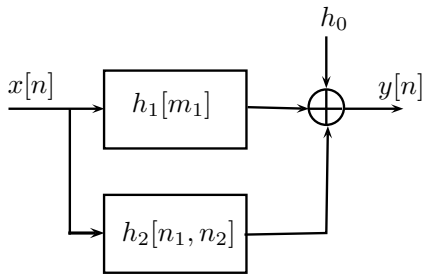



**Fig. 2.2.** Quadratic filter as a parallel combination of 3 components

Eq. 2.10 is understood as the output of linear filter of kernel $\mathsf{H}_2$ operating on

$$\mathsf{X}_2[n] = \mathsf{X}_1[n] \otimes \mathsf{X}_1[n] \tag{2.11}$$

where $\otimes$ represents Kronecker product. Quadratic nonlinearities are predominant in image generation and processing mechanisms. Employing two dimensional quadratic filters can lead to improved performance. Sec. 2.5 details the theory of 2-D quadratic filters.

## 2.5 Two Dimensional Quadratic Systems

The two dimensional quadratic filter is governed by the equation

$$y[n_1,n_2] = \sum_{m_{11}=0}^{N_1-1}\sum_{m_{12}=0}^{N_2-1}\sum_{m_{21}=0}^{N_1-1}\sum_{m_{22}=0}^{N_2-1} h_2[m_{11},m_{12},m_{21},m_{22}] \times x[n_1-m_{11},n_2-m_{12}]x[n_1-m_{21},n_2-m_{22}] \quad (2.12)$$

Out of the four indices $m_{11}, m_{12}, m_{21}, m_{22}$ of the kernel $h_2$, two stem from the quadratic nature of the kernel and the remaining two denote the two dimensions of the signal processed. Eq. 2.12 is represented in the matrix form as

$$y[n_1,n_2] = \mathsf{X}^T[n_1,n_2]\mathsf{H}_2\mathsf{X}[n_1,n_2] \quad (2.13)$$

where

$$\mathsf{X}[n_1,n_2] = \begin{bmatrix} x[n_1,n_2] \\ x[n_1-1,n_2] \\ \vdots \\ x[n_1-N_1+1,n_2] \\ \vdots \\ x[n_1,n_2-N_2+1] \\ x[n_1-1,n_2-N_2+1] \\ \vdots \\ x[n_1-N_1+1,n_2-N_2+1] \end{bmatrix} \quad (2.14)$$

The quadratic kernel $\mathsf{H}_2$ has $N_1N_2 \times N_1N_2$ elements and each element consists of $N_2^2$ sub-matrices $\mathsf{H}(i,j)$ with $N_1 \times N_2$ elements given as

$$\mathsf{H}_2 = \begin{bmatrix} \mathsf{H}[0,0] & \mathsf{H}[0,1] & \cdots & \mathsf{H}[0,N_2-1] \\ \mathsf{H}[1,0] & \mathsf{H}[1,1] & \cdots & \mathsf{H}[1,N_2-1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathsf{H}[N_2-1,0] & \mathsf{H}[N_2-1,1] & \cdots & \mathsf{H}[N_2-1,N_2-1] \end{bmatrix} \tag{2.15}$$

where each sub-matrix $\mathsf{H}[i,j]$ is given by

$$\mathsf{H}(i,j) = \begin{bmatrix} h[0,i,0,j] & h[0,i,1,j] & \cdots & h[0,i,N_1-1,j] \\ h[1,i,0,j] & h[1,i,1,j] & \cdots & h[1,i,N_1-1,j] \\ \vdots & \vdots & \ddots & \vdots \\ h[N_1-1,i,0,j] & h[N_1-1,i,1,j] & \cdots & h[N_1-1,i,N_1-1,j] \end{bmatrix} \tag{2.16}$$

The design of the quadratic filter invariably means the determination of the $N^2$ elements of $\mathsf{H}_2$. This task is tedious as $N^2$ is large enough, but not all the coefficients are independent. The number of independent coefficients is considerably reduced if the isotropy of the kernel matrix is considered as in the following section.

## 2.6 Isotropy of Quadratic Kernel

Isotropy of a one dimensional linear system means the invariance of the filter output with respect of 180° rotation of the filter input. The input to a linear system is

$$\mathsf{X}_1[n] = [\, x[n] \quad x[n-1] \quad \cdots \quad x[n-N+1] \,]^T \tag{2.17}$$

and its image after 180° rotation is

$$\mathsf{X}_1'[n] = [\, x[n-N+1] \quad \cdots \quad x[n-1] \quad x[n] \,]^T \tag{2.18}$$

The output of the isotropic system will be the same for both inputs $\mathsf{X}_1[n]$ and $\mathsf{X}_2[n]$. A class of linear phase FIR filters with impulse response $h_1[n]$ with

$$h_1[n] = h_1[N-1-n] \tag{2.19}$$

can achieve this condition. For a one dimensional quadratic system, the input vector and its image after 180 ° rotation are

$$\mathsf{X}_2[n] = \mathsf{X}_1[n] \otimes \mathsf{X}_1[n] \tag{2.20}$$

$$\mathsf{X}_2^{'}[n] = \mathsf{X}_1^{'}[n] \otimes \mathsf{X}_1^{'}[n] \tag{2.21}$$

The condition under which the outputs of a quadratic system for the inputs $\mathsf{X}_2[n]$ and $\mathsf{X}_2^{'}[n]$ are the same is

$$h_2[n_1, n_2] = h_2[N-1-n_1, N-1-n_2] \tag{2.22}$$

### 2.6.1 Isotropy of 2-D Quadratic Systems

The input-output relation for a two dimensional quadratic system is

$$y[n_1, n_2] = Tr[\mathsf{H}_2 \mathsf{X}_2^T[n_1, n_2]] \tag{2.23}$$

where

$$\mathsf{X}_2[n_1, n_2] = \mathsf{X}_1[n_1, n_2] \oplus \mathsf{X}_1[n_1, n_2] \tag{2.24}$$

Isotropy requires that the output is the same for 90 °, 180 ° and 270 ° rotations of $\mathsf{X}_2[n_1, n_2]$. Isotropy leads to less number of independent coefficients and ease of design [Mathews and Sicuranza 2000].

Two dimensional filters with a small plane of support is ideal for spatial filtering. A filter of $3 \times 3$ mask is considered, with $N = 9$. The number of independent coefficients are reduced from 81 to 45, on application of the symmetry of the $\mathsf{H}_2$ kernel. On application of the isotropy constraints, it reduces to 11 [Ramponi 1990].

## 2.7 Gist of Observations

This chapter presents the limitations of LTI systems and the signal processing needs that necessitate the use of quadratic systems. Volterra series, an extension of Taylor series, is presented as a suitable power series expansion to model quadratic systems. Two dimensional quadratic systems and the properties of Volterra quadratic kernels such as symmetry and isotropy, which will be later used in subsequent designs, are reviewed. The principal distinction when working with quadratic filters is that in the case of 1-D and 2-D linear time invariant systems, the linear convolutions as in Eq. 2.1 and Eq. 2.2 translate into multiplication of Fourier transforms of the input and impulse response functions in the frequency domain i.e. $Y(\omega) = H(\omega)X(\omega)$. The latter statement implies that the frequencies at the output of LTI systems contain the input frequencies or a subset therein, depending on the shape of $H(\omega)$ where $H(\omega)$ is the Fourier transform of $h[n]$, *but never anything outside the input frequency band.* Such a convenient situation does not exist when polynomial systems such as $y[n] = x^2[n]$ is considered. With such a system, if $x[n]$ is band limited in $[-\omega_c, \omega_c]$, then the spectrum of $y[n]$ can spread from $[-2\omega_c, 2\omega_c]$. Not only that the convenient relationship as proposed by Eq. 2.1 or by its frequency domain counterpart does not exist in the case of quadratic systems. Besides, one has to worry about spectral aliasing not only at the input side but at the output side also. If one takes a closer look at systems whose input and output are related by polynomial equations such as the case just discussed, one can observe that there are equivalences with linear systems in the time domain but there is no exact equivalent of the frequency domain. This imposes a great challenge in the implementation of polynomial systems compared with linear systems that are implemented using frequency domain methods that rely on computationally efficient fast Fourier transform algorithms.

## 2.8 Motivation for Research

The research in quadratic systems is motivated by the need to model the effects of polynomial components present in speech and image signals. It is estimated that quadratic systems can outperform LTI systems and other nonlinear filters in the tasks of edge detection, impulsive noise removal and statistical prediction. The design methodology and scheme of implementation need to be selected for each application. Based on the review of literature and the requirements in polynomial signal processing in three application areas, the scheme of work and methodology have been established in tune with the objectives. Chapter 3 is devoted for explaining the scheme of work and methodology.

Chapter 3

# Scheme of Work and Methodology

## Contents

## 3.1 Overview

Polynomial filters, especially the quadratic systems, discussed in chapter 2 find extensive applications in speech and image processing, two areas where effects of nonlinearity are inherent and predominant. The following aspects of quadratic systems are considered below.

- Mathematical formulation of quadratic systems and the various actual and approximate representations of its kernel matrix $\mathsf{H}_2$.
- Isotropy of quadratic kernel $\mathsf{H}_2$ that can result in the saving of computational complexity stemming from Kronecker products.
- Design methodology to find the quadratic kernel matrix $\mathsf{H}_2$.
- Implementation strategies for $\mathsf{H}_2$.

Once the mathematical formulation, design and implementation strategy of quadratic filters are understood, it is imperative to consider specific applications where the features of such filters are put to better use. These fields are identified as

- Quadratic edge detection in biometric and medical images
- Removal of impulsive noise using quadratic filter
- Quadratic predictor for speech signals.

The first work is motivated by the improved edge crispness and noise invulnerability of two dimensional quadratic systems which makes them ideal edge detection filters. Teager and general quadratic edge detection filters are designed and tested for the following applications.

- Detection of retinal microaneurysms due to diabetic retinopathy.
- Enhancement of latent fingerprints in noisy background.

Removal of impulsive noise is an important problem in image processing. One major area where images are corrupted by impulsive noise is that of magnetic resonance imaging. It is proposed that a quadratic Volterra filter can remove impulsive noise much better than a conventional image filter like median or Gaussian filter. So impulsive noise removal is the second area of application of quadratic filters.

The third application area for which quadratic filters are designed and tested is that of nonlinear prediction. The motivation for research is the need for accounting the polynomial nonlinearities in speech signals, due to multiple reflections in the vocal tract. This model is incorporated in the design of a speech predictor. Such a quadratic predictor is included in a differential pulse code modulation system for improved mean square error between transmitted and received speech signals sent over an additive white Gaussian noise (AWGN) channel. The broad categorization of work is shown in Fig. 3.1. The impact of work in each category is as outlined in the subsequent sections.

### 3.1.1 Why the Three Areas?

The above three areas are chosen with the surmise that effects of polynomial nonlinearities are predominant in them and a well designed quadratic filter can outperform conventional filters. The specific requirements and constraints in each area are discussed as follows.

#### Prediction

Prediction involves in estimating the future value of a random process, often realized using linear FIR filters. In the present research, quadratic Volterra filters are used as speech signal predictors, instead of FIR filters, with remarkable improvement in performance. The work is justified by the fact that speech generation is a nonlinear process

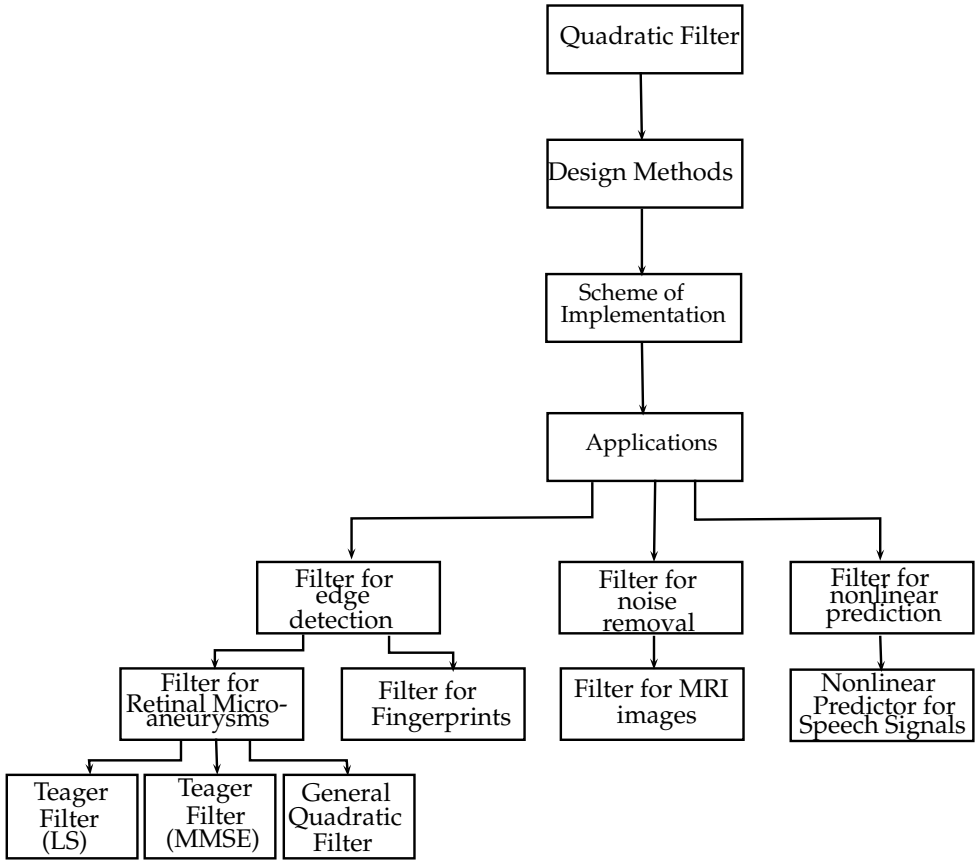



**Fig. 3.1.** Scheme of research

and the inherent polynomial components are modeled better using a quadratic predictor. This quadratic predictor is employed in a differential pulse code modulation system with reduced mean square error between the transmitted and received signals.

### Edge Detection

In image processing and machine vision, isolation of edges is a key process and many filters such as Canny, Sobel, Laplacian, Laplacian of Gaussian(LoG) have been used conventionally for this end. These filters separate edges by computing spatial gradients, but they perform poorly in presence of additive noise. It is proposed that quadratic filters can detect the edges with greater resolution than conventional edge detectors even in presence of strong additive noise. Besides, edges in images are discontinuities in space which are approximated better with nonlinear functions and hence the motivation for employing quadratic systems for edge detection in presence of noise.

### Noise Removal

Any unwanted signal that gets added with the desired signal is termed noise, the removal of which is a fundamental filtering problem. In communication systems where a large number of noise sources get added, the resulting noise is Gaussian in statistics and well established filtering methods like matched filtering are employed for noise removal. But in image processing the noise encountered is impulsive in behaviour, such as salt and pepper noise, the removal of which is done using nonlinear filters like order statistic filters. Gaussian filter also removes impulsive noise though it slightly blurs the image. Apart from these, quadratic filters are proposed for greater improvement in signal to noise ratio and better preservation of edges. The work is motivated

by the noise invulnerability of quadratic edge detection filters.

## 3.2 Filters for Edge Detection

Edges are discontinuities in brightness in images due to peripheries of objects, changes in illumination, texture etc. and manifest as sharp changes in pixel values. These changes in values can be gradual as in Fig. 3.2 or abrupt as in Fig. 3.3. Detection of edges is an important

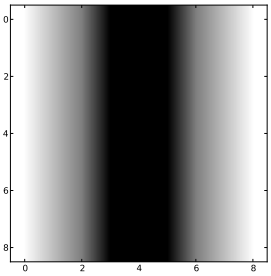



**Fig. 3.2.** A blunt synthetic edge

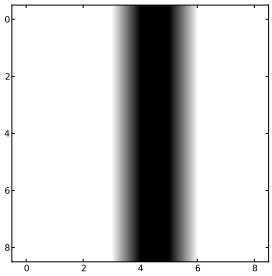



**Fig. 3.3.** A sharp synthetic edge

image processing operation that is useful in periphery detection and machine vision[Pratt 2001],[Weeks 2005]. The applications of edge detection include tumour localization in medical images for automated surgery, geographic localization in satellite images, periphery detection of objects for robotic vision etc. Conventionally, spatial gradient based detectors are used to locate the edges in images. The popular filters under this category are Laplacian, Laplacian of Gaussian(LoG), Canny etc. The major drawback with these methods is that they are very sensitive to additive noise. Noise introduces false edges in an image and the above mentioned filters perform poorly in presence of noise. But quadratic edge detection filters are less sensitive to noise and

hence the motivation for the design and implementation of quadratic filter kernels for the following image processing applications. They are

- quadratic edge detection filter for detecting microanneurysms in retinal images due to diabetic retinopathy
- quadratic edge detection filter for the enhancement of noisy fingerprints

The design and implementation of each of these are explained in the chapters from 5 to 6 in part II.

## 3.3 Filters for Noise Removal

Noise is any unwanted signal that gets added with desired signals and its effect is to obscure the latter. The presence of noise is unavoidable and all one can do is minimizing it. The demarcation, as to what is signal and what is noise, is relative to the user. For example, for a pulmonologist who listens to the sound from a patient's chest the rhythmic breathing sound is the signal and the heart beat in the backgound is the noise. The reverse is the case for a cardiologist. The sources of noise can be external to the system or from within the system that processes the signal. External noise sources affect the signal mostly during transmission through channels. The channels can be dedicated communication media such as wireless and wireline electromagnetic channels, optical fibres etc. or storage channels like magnetic hard disk etc. On transmission through communication channels, a large number of noise sources of varying statistics get added, and by the central limit theorem, the total noise follows Gaussian statistics. Besides external noise, there are inherent noise sources within systems. Such are Johnson noise due to thermally generated carriers in amplifiers, shot noise in conductors etc. One chief figure of merit, when working

in presence of noise, is the signal to noise ratio(SNR) which degrades as the signal progresses through the system because of the presence of internal noise sources. Isolation of desired signals from noise is a fundamental engineering problem to which linear system theory offers well established solutions and methods especially when the noise has Gaussian statistics. But linear systems fail when it comes to removing impulsive noise. Such a noise that is predominant in wireless communication signals, medical images etc., is removed using quadratic filters. Fig. 3.4 shows the original lena image and Fig. 3.5 shows the effect of added impulsive noise of power 100 on it.


**Fig. 3.4.** Lena image

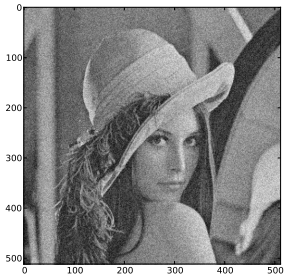



**Fig. 3.5.** Image with noise

Unlike images generated by digital cameras, images generated by medical imaging systems such as MRI machine are largely corrupted by impulsive noise as the acquisition of image happens under strong and rapidly varying magnetic fields. In the present work, a two dimensional filter kernel is developed and implemented for removing impulsive noise from magnetic resonance image(MRI) signals. The principal advantages in using quadratic filter are its edge and structure preservation features and noise invulnerability. The details of work done on this class of filter are explained in part III.

## 3.4 Filters for Nonlinear Prediction

Entropy, information and randomness are manifestations of the same entity, measurement of which is a fundamental problem in information theory, for which solutions were offered by Chaitin and Claude Shannon. According to Shannon,

> "we have full knowledge about the past but cannot control it; we can control future but have no knowledge of it."

Lack of knowledge about the future motivated the research in prediction. One is always interested to know what will happen in future or at least what is the most probable thing that will happen in future. Theories of estimation and prediction stem from this interest and Nobert Wiener, a lightning brained mathematician, undertook this problem and predicted the trajectory of a bomb during the second world war. The later developments in the two areas are attributed to him.

Linear prediction is an important statistical signal processing operation that involves in predicting the future value of a random variable based on the past samples. This finds applications in speech coding, image sequence prediction, trajectory prediction, market prediction etc. The performance of predictors will be further improved if polynomial predictors that can model the nonlinearities in speech and image signals are employed in the differential coding systems. Quadratic predictors are designed and implemented for the following applications.

- Lattice type quadratic predictor for Gaussian signals
- Optimization based quadratic predictor for the differential coding and decoding of speech signals

These quadratic filters are contrasted against the linear predictor based on Levinson-Durbin recursion for testing the performance parameters.

The research work and the results obtained in differential PCM systems employing quadratic predictors are detailed in part IV.

## 3.5 Scope of Work

The research work in the areas of nonlinear prediction, edge detection and noise removal are targeted for various applications. The scope of work in various areas are outlined in the subsequent sections.

### 3.5.1 Strategy for Design and Implementation

The methods proposed for the design of quadratic systems lack generality. But two major methods are considered and the pros and cons of both are analyzed before deciding on one. Once the design for a specific application is accomplished, a computationally efficient method realization should be selected. Several methods are compared before matrix decomposition is selected.

### 3.5.2 Quadratic Edge Detection Filter

Two dimensional Quadratic filters are proposed for detecting edges in images with improved edge resolution and noise invulnerability. Such features are desirable for applications like

- enhancement of noisy fingerprints
- detection of retinal microaneurysms

The first application is useful in enhancing crime scene fingerprints that are unclear or buried in noise. It is a pre - processing stage before the prints are subjected to identification. Quadratic edge detection

filter that preserves the structure of image even with additive noise is used in an unsharp masking scheme for fingerprint enhancement.

The periphery of retinal microaneurysms typically $100\mu m$ dimension due to type - I and type - II diabetes are enhanced using a variation of quadratic filters viz. Teager filters, which track energy localizations and consequently the edges in the input image. A general quadratic filter can also be employed for the enhancement of retinal microaneurysms. The precise detection of periphery aids automatic surgery of microaneurysms in the initial stage of diabetic retinopathy. It forms the second application area of quadratic edge detection filter.

### 3.5.3 Modified DPCM System with Quadratic Predictor

The polynomial components in speech signals, due to multiple reflection in the vocal tract during speech generation, are not accounted for in a linear predictor and so it yields a large prediction error. These components are modeled using Volterra power series. A speech predictor based on Volterra quadratic system is proposed to yield smaller mean square error between actual speech signal and its predicted value. The quadratic predictor is used to replace the linear predictor in a differential pulse code modulation system, with improved performance.

### 3.5.4 Removal of Impulsive Noise from MRI Signals

Unlike in communication systems, the noise present in images are impulsive in nature, making filtering difficult. One key area where strong impulsive noise is present, and its removal is a critical operation, is

the generation of magnetic resonance imaging (MRI) signal. In MRI, nuclear resonances due to rapidly changing magnetic fields are transformed into images of body part under study. The strong variations in magnetic fields add impulsive noise with the resulting image which need to be removed. Conventionally, order statistic filters such as mean or median filters are employed for this task with unavoidable blurring of edges. A quadratic filter is used for the removal of impulsive noise from MRI images with greater signal to noise ratio and edge crispness than linear filters and conventional nonlinear filters.

## 3.6 Summary

This chapter details the scheme of research work adopted in the design and implementation of quadratic filters and their realizations for three applications. At the onset, quadratic filter is selected as a potential choice for nonlinear signal and image processing. Edge detection, noise removal and prediction are identified as areas where quadratic filtering could exploit more than that exploited by linear or other nonlinear filters in terms of performance parameters. The division of work under each category is also presented. The forthcoming parts in the thesis detail the work on quadratic system for the three areas.

Chapter 4

# Design and Implementation Strategies for Quadratic Systems

## Contents

## 4.1 Introduction

In the last chapter, the potential application areas, where quadratic systems can contribute more than conventional systems, are identified and the objectives are laid out. Now, the strategies for the design of quadratic systems are to be sought. There is no general methodology for the design of quadratic systems unlike in the case with LTI systems. Mostly, the methods are heuristic and application dependent but bestows the designer with the power to customize the filter to suit the application. The complexity of design is considerably reduced if the symmetry and isotropy of the quadratic kernel are exploited. Two methods have been proposed for finding the quadratic kernel, although they lack generality. These methods are detailed in Sec. 4.2. Once the quadratic kernel is designed, it is imperative to look for a proper strategy of implementation. Since the complexity of Kronecker products increases sharply with the size of the input array, a direct implementation is impossible. Computationally efficient tools like FFT, that are useful with LTI systems, are of little use due to the absence of frequency domain in the case of quadratic systems. Realizations based on distributed arithmetic are possible[White 1989]. In this case, however, the multiplications are converted into serial additions and do not offer considerable computational throughput in the case of quadratic systems. Approximate implementations of the kernel are possible by resorting to suitable matrix decompositions [Gantmacher 1960]. These approximate methods are contrasted with other methods in Sec. 4.3.5. General matrix decomposition, the LU decomposition and the singular value decomposition are possible for the kernel matrix under the category of approximate implementations. Comparison of matrix decomposition methods is made in Sec. 4.3.5 to select singular value decomposition(SVD) method. Approximation of kernel matrix by SVD method is used throughout the research work for implementing both 1-D and 2-D quadratic systems.

## 4.2 Design Methodology

Design of quadratic filters generally involves in finding the kernel $\mathsf{H}_2$ for which two methods [Mathews and Sicuranza 2000] are proposed as

- Bi-impulse response method
- Optimization method

### Bi-impulse Response Method

Two dimensional quadratic system with a small plane of support is useful in image processing. Such a system is characterized by its bi-impulse response which is the output of the system for two unit impulses located at different positions in the plane of filter support[Ramponi 1990]. If the quadratic system is excited by the sum of two impulses $\delta_a[n_1 - k_{11}, n_2 - k_{12}]$ and $\delta_b[n_1 - k_{21}, n_2 - k_{22}]$, then the response is

$$y_q[n_1, n_2] = h_2[n_1 - k_{11}, n_2 - k_{12}, n_1 - k_{21}, n_2 - k_{22}] \tag{4.1}$$

Fig. 4.1 shows the location of the four impulses that are used to determine a $9 \times 9$ $\mathsf{H}_2$ kernel.

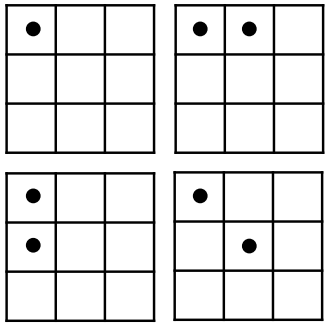


**Fig. 4.1.** Location of bi-impulses in Table 4.1

**Table 4.1:** Indices of coefficients of $\mathsf{H}_2$ determined by four impulses

| Sl. No. | First Impulse | Second Impulse | Elements of $\mathsf{H}_2$ identified |
|---|---|---|---|
| 1 | $\delta_a[n_1, n_2]$ | $\delta_b[n_1, n_2]$ | 11 22 33 44 55 66 77 88 99 |
| 2 | $\delta_a[n_1, n_2]$ | $\delta_b[n_1, n_2 - 1]$ | 12 23 45 56 78 89 |
| 3 | $\delta_a[n_1, n_2]$ | $\delta_b[n_1 - 1, n_2]$ | 14 25 36 47 58 69 |
| 4 | $\delta_a[n_1, n_2]$ | $\delta_b[n_1 - 1, n_2 - 1]$ | 15 26 48 59 |

Table 4.1 shows the indices of the kernel $\mathsf{H}_2$ that are ascertained with the above four sets of impulses. The impulses in the fourth row gives the kernel coefficients $\mathsf{H}_2[1, 5]$, $\mathsf{H}_2[2, 6]$, $\mathsf{H}_2[4, 8]$ and $\mathsf{H}_2[5, 9]$.

The responses for all impulses corresponding to the independent coefficients are used to ascertain all the elements in the kernel $\mathsf{H}_2$. Symmetry and isotropy of the filter kernel are further exploited to narrow the search for filter coefficients.

## Optimization Method

Design by bi-impulse responses lacks generality and it is difficult to impose constraints on $\mathsf{H}_2$. As a better alternative, the design of $\mathsf{H}_2$ is formulated as an optimization problem, based on some constraint function, that minimizes some cost function. The structure of the filter derived, the constraint function and the cost function are all dependent on the specific application that the designer is interested in[Ramponi and Ukowich 1987]. The design problem is formulated as

the identification of the vector $\mathsf{h}^0$ such that

$$\mathrm{f}(\mathsf{h}^0) = \min_{\mathsf{h}\in\mathcal{D}} \mathrm{f}(\mathsf{h}) \tag{4.2}$$

where $\mathsf{h}$ is the $k$-dimensional design variable. $\mathcal{D}$ is the set of all vectors $\mathsf{h}$ that satisfies the constraint function $g(\mathsf{h}) = 0$. The cost function f(.) is a real non-negative function. Sometimes, it is difficult to obtain a theoretical formulation of an objective function. This difficulty is surmounted by training the filter with known input and desired output images. Sometimes, synthetic images with characteristics similar to those of practical images are used at the input and output to ascertain the quadratic filter kernel. Typically, signal to noise ratio(SNR), mean square error(MSE) between a desired image and the filter output, edge crispness etc. are used as cost functions. Constraints are imposed, exploiting the symmetry and isotropy of the quadratic kernel, as explained in Sec. 2.6. Conjugate gradient algorithm is preferred as it offers fast convergence[Fletcher and Powell 1963], [Brent 1973].

**Comparison between Design Methods**

Design of quadratic filters invariably means the determination of the kernel $\mathsf{H}_2$ for the desired application. A comparison of optimization method with bi-impulse response method is as shown in Table 4.2. Out of the two methods outlined above, optimization is used in this work as it can be customized to suit the various applications.

## 4.3 Implementation of Quadratic Filters

Once the strategy for the design of quadratic filters is understood, it is imperative to look into the aspects of their realizations. The complexity of the structure is decided by the number of multiplications which is far more than that with linear systems. The methods of realization

**Table 4.2:** Comparison between design methods

| Sl. No. | Attribute | Bi-impulse response method of design | Optimization method of design |
|---|---|---|---|
| 1 | Isotropy | Isotropy constraints can ease the design | Isotropy constraints are not necessary |
| 2 | Customization | The deconvolution of a desired output with bi-impulse to have $H_2$ for a specific application is difficult | $H_2$ is found based on the optimization of an objective function. So customization is easy |
| 3 | Strategy | Many constraints are to be imposed for eliminating redundant elements in $H_2$ | Unconstrained optimization is made possible by using Fast convergence algorithms |
| 4 | Flexibility | Inflexible design | Repetitive iterations are possible that can yield global minimum for the objective function. There is added flexibility in selecting suitable objective function. |

of quadratic filters [Mathews and Sicuranza 2000], as listed below, are presented in the subsections 4.3.1, 4.3.2 and 4.3.3 respectively.

- Direct form realization
- Realization based on distributed arithmetic
- Realization based on matrix decomposition

### 4.3.1 Direct Form Realization

The homogeneous quadratic components of the Volterra series expansion given by Eq. 2.5 in page 30 can be implemented using *direct-form realization* by means of a nonlinear combiner(computes all necessary products of the input samples), a number of multipliers, and a summing bus. This brute force method that realizes the equation

$$y[n_1, n_2] = \sum_{k_1=0}^{2} \sum_{k_2=0}^{2} h_2[k_1, k_2] x[n_1 - k_1] x[n_2 - k_2] \tag{4.3}$$

as in Figs. 4.2 and 4.3 respectively. The realization shown in Fig. 4.2 uses least number of multipliers and so is computationally simple.

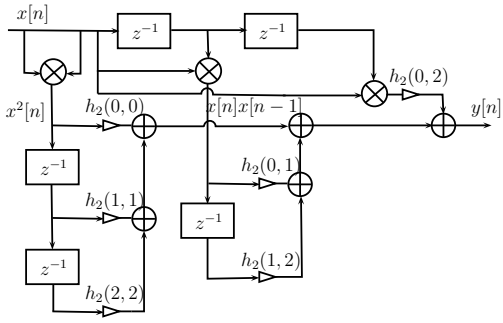



**Fig. 4.2.** Direct form Realization using least number of multipliers

The realization shown in Fig 4.3 uses minimum number of delay elements at the expense of extra multipliers. However, these realizations cannot easily be simplified by approximate representations.

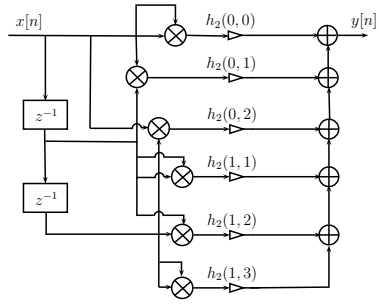



**Fig. 4.3.** Direct form Realization using least number of delay elements

It is impossible to perform direct quadratic filtering on any usable image. The direct form realizations of both one dimensional and two dimensional quadratic filters, as in Eq. 2.10 and Eq. 2.12 in page 31 and in page 32 respectively, are difficult since the computational complexity of Kronecker product increases sharply with the size of the input array. Fig. 4.4 shows the rapid increase in computational complexity of the Kronecker product of a random $N \times N$ array as $N$ rises from 3 to 70. It is impossible to perform direct quadratic filtering on any usable image.

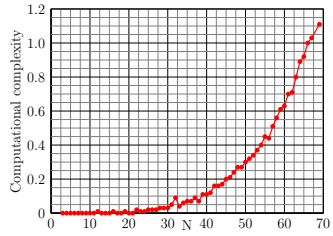



**Fig. 4.4.** Computational complexity arising from Kronecker product of $N \times N$ array with itself

### 4.3.2 Structures Based on Distributed Arithmetic

The concept behind the strategy of implementation is to *distribute* the multiplication and addition operations at the bit level. Assume that the input signal is normalized so that $|x[n]| \leq 1$ for all $n$. In order to perform the multiplication and addition operations at the bit level, each input sample is coded by means of logical variables $x_b[n]$ that assume binary values such that

$$x[n] = \sum_{b=1}^{B} p[b] x_b[n] \tag{4.4}$$

where $B$ is the word length used and weights $p[b]$ depends on the binary code employed. For the two's-complement code,

$$p[1] = -1 \text{ and } p[b] = 2^{-b+1};\ b = 2, \ldots, B \tag{4.5}$$

and $x_b[n]$ is either 1 or 0. The offset - binary code requires that $x_b[n] = \pm 1$ and

$$p[b] = 2^{-b};\ b = 1, \cdots, B \tag{4.6}$$

The definition in Eq. 4.4 is extended to the past $N-1$ input samples and the input vector of Eq. 4.5 is expressed in the form

$$\mathsf{X}_1[n] = \mathsf{Q}_1[n]\mathsf{P}_1 \tag{4.7}$$

where $\mathsf{P}_1$ is a $B$-element vector of the binary weights given by

$$\mathsf{P}_1 = \begin{bmatrix} p[1] & p[2] & \ldots & p[B] \end{bmatrix}^T \tag{4.8}$$

and $\mathsf{Q}_1(n)$ is an $N \times B$-element matrix obtained by expanding each component of the vector $\mathsf{X}_1(n)$ in its elementary bits as

$$\mathsf{Q}_1[n] = \begin{bmatrix} x_1[n] & x_2[n] & \dots & x_B[n] \\ x_1[n-1] & x_2[n-1] & \dots & x_B[n-1] \\ x_1[n-2] & x_2[n-2] & \dots & x_B[n-2] \\ \vdots & \vdots & \ddots & \vdots \\ x_1[n-N+1] & x_2[n-N+1] & \dots & x_B[n-N+1] \end{bmatrix} \quad (4.9)$$

Consequently, the output of a linear FIR filter can be expressed using the vector notation

$$y_1[n] = \mathsf{X}_1^T[n]\mathsf{H}_1 = \mathsf{P}_1^T\mathsf{Q}_1^T[n]\mathsf{H}_1 \quad (4.10)$$

where $\mathsf{H}_1$ is the impulse response of filter in vector form.

Calculation of the product $\mathsf{P}_1^T(\mathsf{Q}_1^T[n]\mathsf{H}_1)$, that is devoid of any multiplication, involves only bit-shifting, additions and subtractions. It is required to substitute Eq. 4.7 in Eq. 2.11 in page 31 to derive a distributed arithmetic realization of a homogeneous quadratic filter as given below.

$$\begin{aligned} \mathsf{X}_2[n] &= \mathsf{X}_1[n] \otimes \mathsf{X}_1[n] \\ &= \mathsf{Q}_1[n]\mathsf{P}_1 \otimes \mathsf{Q}_1[n]\mathsf{P}_1 \\ &= [\,\mathsf{Q}_1[n] \otimes \mathsf{Q}_1[n]\,]\,[\,\mathsf{P}_1 \otimes \mathsf{P}_1] \\ &= \mathsf{Q}_2[n]\mathsf{P}_2 \end{aligned} \quad (4.11)$$

where $\mathsf{Q}_2[n] = \mathsf{Q}_1[n] \otimes \mathsf{Q}_1[n]$ is an $N^2 \times B^2$-element matrix and $\mathsf{P}_2 = \mathsf{P}_1 \otimes \mathsf{P}_1$ is a $B^2$-element vector. The output of the homogeneous quadratic filter is expressed in terms of $\mathsf{P}$ and $\mathsf{Q}$ by the equation

$$y_2[n] = \mathsf{X}_2^T[n]\mathsf{H}_2 = \mathsf{P}_2^T\mathsf{Q}_2^T[n]\mathsf{H}_2 \quad (4.12)$$

The symmetry property of the Volterra kernels is used to reduce the implementation complexity by defining a small vector $\mathsf{H_{2r}}$ containing only $N_2 = N(N+1)/2$ independent coefficients in the quadratic kernel, and a corresponding reduced-size input vector $\mathsf{X_{2r}}[n]$.

$$y_2[n] = \mathsf{X}_{2r}^T[n]\mathsf{H_{2r}} = \mathsf{P}_2^T\mathsf{Q}_{2r}^T[n]\mathsf{H_{2r}} \tag{4.13}$$

where $\mathsf{Q_{2r}}[n]$ is an $N_2 \times B^2$-element matrix obtained by expanding the $B^2$-elementary bits in the vector $\mathsf{X_{2r}}[n]$ [Mathews and Sicuranza 2000]. The realization is as shown in Fig. 4.5.

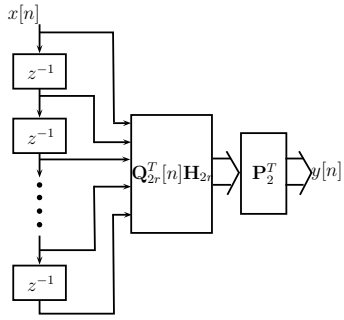



**Fig. 4.5.** Structure based on distributed arithmetic

### 4.3.3 Implementation of One Dimensional Quadratic Kernel by Matrix Decomposition

Any symmetric $N \times N$ matrix $\mathsf{H_2}$ of rank $r$ is decomposed into a sum of $r$ number of rank - 1 matrices as an approximation $\tilde{\mathsf{H}}_2$, expressed as

$$\tilde{\mathsf{H}}_2 = \sum_{i=1}^{r} q_i \mathsf{R_i}\mathsf{R_i}^T \tag{4.14}$$

where $\{ q_i; i = 1, 2, 3, \ldots, r\}$ are scalars and $\{ \mathsf{R_i}; i = 1, 2, 3, \cdots, r\}$ are N element vectors. Substituting Eq. 4.14 in Eq. 2.10, the quadratic term of output becomes

$$y_q[n] = \sum_{i=1}^{r} q_i [\, \mathsf{X_1}^T[n] \mathsf{R_i} \,] [\, \mathsf{R_i}^T \mathsf{X_1}[n]] \tag{4.15}$$

$$y_q[n] = \sum_{i=1}^{r} q_i y_i^2[n] \tag{4.16}$$

where

$$y_i[n] = \mathsf{X}_1^T[n] \mathsf{R_i} = \mathsf{R_i^T X_1}[n] \tag{4.17}$$

is the output of a linear filter with impulse response equal to the coefficient vector $\mathsf{R_i}$. The realization is as shown in Fig. 4.6.

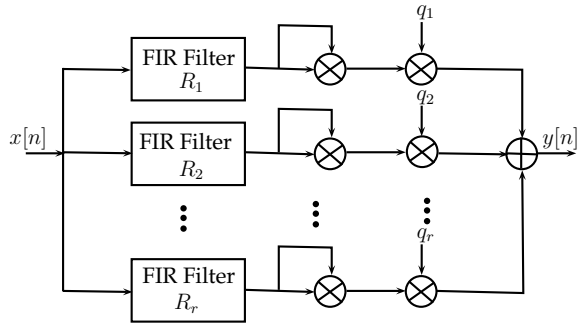



**Fig. 4.6.** Realization of $\tilde{\mathsf{H}_2}$ by general matrix decomposition

The input is applied to the FIR filters with impulse responses $\mathsf{R_i}$ and each output is squared and then scaled by $q_i$. General matrix decomposition does not result in compression of the kernel matrix $\mathsf{H_2}$. This means that not every $q_i$ is insignificant enough and not every $\mathsf{R_i}$ contains leading zeros, resulting in meager computational savings.

### LU Decomposition

A more efficient realization is by LU decomposition in which a matrix is represented as the product of upper and lower triangular matrices. The matrix $\mathsf{H}_2$ is approximated, by LU method, as $\tilde{\mathsf{H}_2}$ in the form of a weighted sum in the equation below.

$$\tilde{\mathsf{H}_2} = \sum_{i=1}^{r} d_i \mathsf{L_i} \mathsf{L_i}^T \tag{4.18}$$

where $d_i$ is scalar multiplier and vector $\mathsf{L_i}$ has $i-1$ leading zeros. Thus LU decomposition results in less complexity than general matrix decomposition.

### Singular Value Decomposition

A more convenient decomposition of an array or matrix is the singular value decomposition that approximates a matrix as a linear combination of singular vectors, scaled by the respective eigenvalues. It is observed as a method for transforming correlated values in the $\mathsf{H}_2$ matrix into a set of uncorrelated values that exposes the relationships among the original values. This method identifies and orders the dimensions along which the elements exhibit the most variation. Singular value decomposition is a popular method of image compression and is used to derive approximate realizations when rank of the kernel matrix $\mathsf{H}_2$ is smaller than the memory span. Here $\mathsf{H}_2$ is approximated as

$$\tilde{\mathsf{H}_2} = \sum_{i=1}^{r} \lambda_i \mathsf{S_i} \mathsf{S_i}^T \tag{4.19}$$

where $\lambda_i$ are the singular values of $\mathsf{H}_2$ and $\mathsf{S_i}$ are the corresponding orthonormal singular vectors. The best rank-$\rho$ approximation is

$$\tilde{\mathsf{H}_2} = \sum_{i=1}^{\rho} \lambda_i \mathsf{S_i} \mathsf{S_i}^T \tag{4.20}$$

where $\{\lambda_i; i = 1, 2, 3 \ldots, \rho\}$ are the largest $\rho$ singular values of the kernel $\mathsf{H}_2$. The output of the filter is

$$\begin{aligned} y[n] &= \mathsf{X}^T[n]\mathsf{H}_2\mathsf{X}[n] \\ &\approx \mathsf{X}^T[n] \sum_{i=1}^{\rho} \lambda_i \mathsf{S_i} \mathsf{S_i}^T \mathsf{X}[n] \\ &= \sum_{i=1}^{\rho} \lambda_i [\, \mathsf{X}^T[n]\mathsf{S_i}\,][\,\mathsf{S_i}^T\mathsf{X}[n]\,] = \sum_{i=1}^{\rho} \lambda_i y_i^2[n] \qquad (4.21) \\ & \qquad (4.22) \end{aligned}$$

The parallel-cascade structure is as shown in Fig. 4.7.

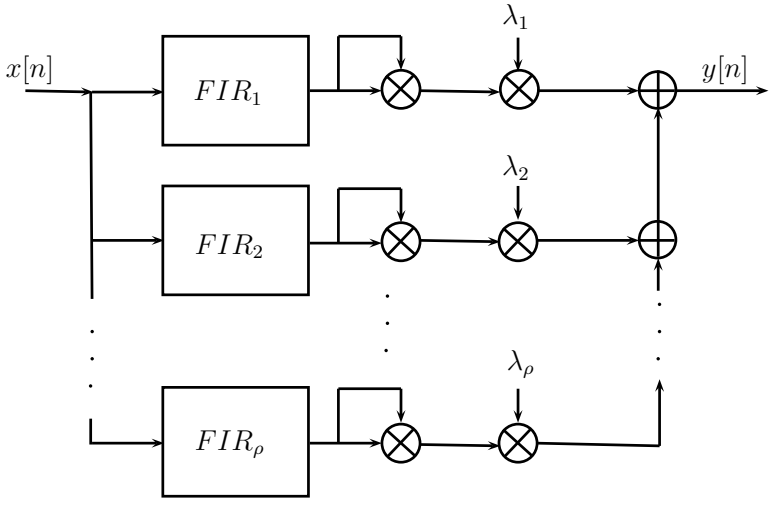



**Fig. 4.7.** Realization of $\tilde{H}_2$ of by singular value decomposition

The Frobenius norm of the error matrix $\mathsf{H}_2 - \tilde{\mathsf{H}_2}$ is given as[Gantmacher 1960]

$$||\mathsf{H}_2 - \tilde{\mathsf{H}_2}||_F = \sqrt{\sum_{i=1}^{\rho} \lambda_i^2} \tag{4.23}$$

Often, the implementation with $\tilde{\mathsf{H}_2}$ results in substantially smaller number of coefficients. Further saving in computational complexity will be achieved if LU decomposition is performed on $\tilde{\mathsf{H}_2}$ as well.

### 4.3.4 Implementation of Two Dimensional Quadratic Kernel by Matrix Decomposition

The idea of decomposition of $\mathsf{H}_2$ is extended to two dimensional systems. Singular value decomposition identifies the elements of $\mathsf{H}_2$ along the direction of maximum variation. Once this direction is ascertained, one can find the approximation for the original elements in $\mathsf{H}_2$ using fewer dimensions. Thus singular value decomposition is performed over $\mathsf{H}_2$ in Eq. 2.15. Upon singular value decomposition of the kernel matrix $\mathsf{H}_2$, it yields an approximation $\tilde{\mathsf{H}_2}$ again as a weighted sum

$$\tilde{\mathsf{H}_2} = \sum_{i=1}^{\rho} \lambda_i \mathsf{S_i} \mathsf{S_i}^T \tag{4.24}$$

where $\lambda_i$ are the eigen vectors and each $\mathsf{S_i}$ is a $N_1 N_2 \times 1$ vector. The order of the filter bank $\rho$ is selected in such a manner that the Frobenius norm $||\mathsf{H}_2 - \tilde{\mathsf{H}_2}||$ is minimum. Each $\mathsf{S_i}$ is re-sized as a $N_1 \times N_2$ FIR image filter that is equivalent to $H(i,j)$ in Eq. 2.16. The outputs of two dimensional FIR filters are squared and a weighted sum with $\lambda_i$ values yields the filter structure given as in Fig. 4.8.

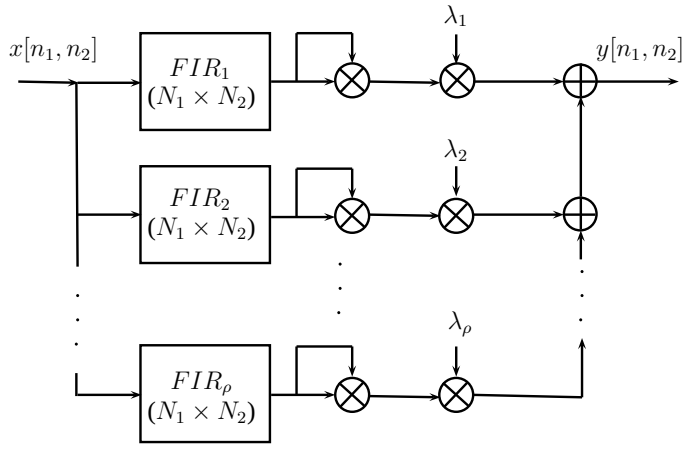



**Fig. 4.8.** Realization of $\tilde{H}_2$ for 2-D quadratic system by singular value decomposition

### 4.3.5 Comparison between Structures

Direct form implementation is a brute force approach and the computational complexity involved is tremendous. Distributed arithmetic structure needs large number of conversions from floating point to binary. The matrix methods result in compact multichannel filters. Comparison between the three methods is presented in Table 4.3.

It is inferred from the table that matrix decomposition methods are better choices of implementation than the other two methods.

#### Comparison between Structures based on Matrix Decomposition

Many matrix decompositions methods are possible for $\mathsf{H}_2$ but singular value decomposition is particularly attractive. This decomposition identifies the path along the principal values of $\mathsf{H}_2$ and yields its best

approximation with fewer dimensions. A comparison in terms of implementation strategy and computational complexity are given in Table 4.4.

**Table 4.3:** Comparison between various structures

| Attribute | Complexity | Method of implementation | Flexibility |
|---|---|---|---|
| Direct form | Large computational complexity | By Kronecker products | Not flexible |
| Distributed arithmetic | Complex with large number of multipliers and adders | Multiplications are converted to additions | The hardware used for performing the conversion, addition and multiplication is not flexible |
| Matrix decomposition | Less complex | Multichannel implementation that approximates the quadratic kernel | Since the principal components of $\mathsf{H}_2$ are isolated, channels are added or dropped depending on the magnitude of eigenvalues, resulting in flexible implementation. |

**Table 4.4:** Comparison between matrix decompositions

| Sl. No. | Structure | Complexity | Implementation |
|---|---|---|---|
| 1 | General matrix decomposition | Full rank approximation is as complex as direct implementation | Parallel cascade with $\rho$ channels[a] |
| 2 | LU decomposition | $\rho$ number of FIR filters and multipliers are needed | Parallel cascade with $\rho$ channels |
| 3 | Singular value decomposition | As eigenvalues decrease rapidly the complexity is substantially reduced as $\mathsf{H}_2$ is expressed with less number of FIR filters | Multichannel implementation with a few FIR filters and squarers |

[a] $\rho$ is the rank of $H_2$

The general matrix decomposition results in a complexity at par with direct implementation, especially when rank of $\mathsf{H}_2$ is high. LU decomposition splits $\mathsf{H}_2$ into lower and upper triangular matrices. Although this decomposition results in FIR filters with increasing number of zeros in their impulse responses, the number of filter channels are not substantially reduced. Singular value decomposition results in rapidly decreasing eigenvalues and so the kernel matrix $\mathsf{H}_2$ is approximated by a few principal values, leading to a computationally efficient structure.

## 4.4 Summary

The chapter presented the need for working with polynomial systems for modeling of mild nonlinearities and the use of Volterra series, rather than Taylor series, for that end. This series enables to add quadratic and higher order nonlinear systems in parallel with LTI systems. In practice, most of the nonlinear effects are modeled by the quadratic term. Higher powers are difficult to implement in hardware and hence the focus on quadratic systems. While bi-impulse response method and optimization method are used for designing quadratic filters, optimization is preferred as it enables application specific design and development. Once the design is done, computationally efficient implementations are to be done for quadratic systems. Matrix decomposition methods, preferably, singular value decomposition is used for approximate implementation of quadratic filters. Three areas of applications are identified, based on the opportunity of exploiting the benefits of quadratic signal processing. The details of the research work in the three areas are discussed in the subsequent chapters.

# Part II

# Quadratic Filters for Edge Detection

# Foreword to Part II

The work done to design and implement quadratic filters for precise edge detection as per the scheme of work shown in Fig. 3.1 is detailed in the chapters from 5 to 6 in part II. Quadratic edge detection filter, used for the enhancement of noisy fingerprints similar to latent fingerprints found in crime scenes, is detailed in chapter 5. This filter has the unique advantage of improved edge crispness and greater noise invulnerability than conventional edge detectors. Besides, it exhibits greater preservation of structure even at high noise levels. It is embedded in an unsharp masking scheme to enhance the contrast of latent fingerprints, deeply lost in background noise.

Enhancement of fundus retinal images, for the exact assessment of the periphery of microaneurysms due to diabetic retinopathy, is essential for automated surgery. Three kinds of quadratic filters are designed and implemented in chapter 6. A two dimensional Teager filter, based on least square method, is designed and tested as in Sec. 6.4.1. Minimization of mean square error is used as a strategy for designing two dimensional Teager filter in Sec. 6.4.2. The frequency domain characteristics of the two are understood and compared. Further, a general quadratic filter is designed based on Powell optimization as given in Sec. 6.4.3. The performance of the three filters are compared among themselves and with conventional nonlinear edge detection filters.

Chapter 5

# Unsharp Masking using Quadratic Filter for the Enhancement of Fingerprints in Noisy Background

## Contents

## 5.1 Introduction

As crime rate increases steadily, fingerprint enhancement and consequent identification always remain a key area of research. Fingerprint is the impression left on a surface by the friction caused by the ridges on a finger or any part of hand and is a unique biometric identifier. The print is composed of dark ridges and light valleys[Maltoni et al. 2003],[Scheibert and Debregeas 2009]. Identification of fingerprints is a key process in access control and in forensic sciences[Arun Ross and Reisman 2003],[Lea and Van 2012]. In the former case, the fingerprints are less noisy and do not require much preprocessing[Jain and Prabhakar 1999]. But the prints taken from crime scenes are blurred and noisy and so require enhancement of ridges to ease identification. The ridges in fingerprints carry significant amount of bio-metric information and they can be enhanced by improving the edge features of the image. Edges in images are formed by discontinuities in spatial, geometrical or photo-metric properties of objects[Pratt 2001]. Although edges are composed of high frequency components, simple high pass filtering does not suffice in detecting and improving edges as it blurs the image. Segmentation of images are done based on texture[Ilea and Whelan 2011],[Hou and Wei 2002],[Shifeng Lia and Zhang 2012] and mathematical morphology. Generally, edges are detected by the computation of the derivative of the image[Vizireanu and Udrea 2009],[Vizireanu and Udrea 2007]. This computation is very noise sensitive as noise appears as false edges in an image. So the chief performance criterion of an edge detector becomes the invulnerability to noise. The spatial gradient based edge detectors [Pratt 2001],[Jain 2003] conventionally employed are

1. Laplace filter

2. Sobel filter

3. Laplacian of Gaussian(LoG)

4. Canny filter

Although Sobel filter has the advantage in speed, it suffers from the lack of edge resolution. The Laplace filter and Canny filter have reasonably good edge resolution but are highly susceptible to noise. LoG filter had good resolution of edges as well as moderate noise invulnerability. But even the LoG does not suffice to enhance edges in images mixed with noise. Under such conditions, polynomial filters perform better in detecting edges[Mitra and Sicuranza 2001] with high enough resolution. Images are formed by nonlinear processes and human vision is inherently nonlinear. So employing polynomial methods for image processing and analysis becomes a natural alternative. Much of the nonlinearities can be modeled by the quadratic term alone and hence the motivation for the design and implementation of quadratic filters. The present work proposes a quadratic filter based on Volterra series for enhancing noisy fingerprints. Although the idea of modeling nonlinearities by power series was proposed a century back by Vito Volterra, the practical realizations and applications were hampered by the large computational complexity. Recently, with increase in computational resources, interest is renewed in developing Volterra systems for signal and image processing with consequent enhancement of features that is otherwise not achievable with linear filters.

## 5.2 Methodology

Enhancement of noisy fingerprints is made possible with unsharp masking scheme in which a scaled version of the edges separated from the fingerprints is added with the noisy prints. The scheme, as given in section 5.3, is proposed for enhancing fingerprint in noisy background. It relies on a quadratic edge detection filter. The flow of work is as

depicted in Fig. 5.1. The first phase of work is in designing the edge detection kernel $\mathsf{H}_2$. The design is based on the minimization of mean square error between a synthetic true edge and its noisy version. The second phase is the computationally efficient implementation of $\mathsf{H}_2$ based on singular value decomposition. These phases are discussed in Section 5.4. In the last phase, testing of the filter is done with standard images corrupted by impulsive and Gaussian noise of different variances and the model is validated in terms of edge preservation in presence of noise. The unsharp masking scheme based on quadratic filter is applied on noisy fingerprints and the performance is compared with schemes employing Gabor, Laplace, Sobel, Canny and LoG filters in terms of the average signal to noise ratio, peak signal to noise ratio and the visibility of ridges. The experiments are detailed in Section 5.5. The results of various experiments are shown in Sec. 5.6.

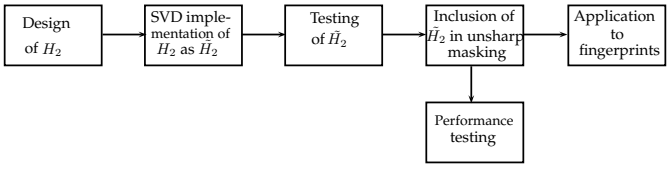



**Fig. 5.1.** Unsharp masking with quadratic filter

## 5.3 Quadratic Unsharp Masking

Unsharp masking is a contrast enhancement scheme[Mitra and Sicuranza 2001] in which a high pass filtered and scaled version of an image is added with itself. The high pass filter enhances the edges and the addition of edges improves the overall contrast of the image. The chief difficulty with this scheme is that the edge detection high pass filters commonly employed are very sensitive to noise since it appears as false edges. It is observed that quadratic edge detectors are very noise immune and have better edge detection characteristics than Laplacian

and LoG filters. The unsharp masking scheme with quadratic edge detection filter is in Fig. 5.2. The edges in the input image $x[n_1, n_2]$ are separated by the quadratic filter. They are then scaled by a factor $\Lambda$ and are added with the input image to yield the enhanced version $\tilde{x}[n_1, n_2]$. The scale factor $\Lambda$ is chosen in such a way that there is improvement in $\tilde{x}[n_1, n_2]$ in respect of visual quality as well as in performance criteria like SNR, PSNR, edge crispness etc.

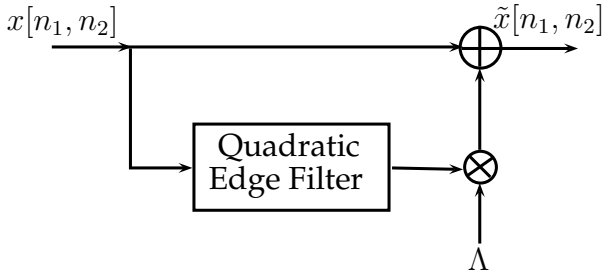



**Fig. 5.2.** Unsharp masking with quadratic filter

## 5.4 Design of Quadratic Edge Detector

It is proposed that a quadratic filter can enhance the edges better than Laplacian or LoG filter. The principal issue in employing a quadratic filter is the identification of its kernel $\mathsf{H_2}$. Powell optimization[Fletcher and Powell 1963] is used for obtaining $\mathsf{H_2}$ as this algorithm has a fast rate of convergence. A synthetic edge with compressed gray scale values added with noise denoted as $x[n_1, n_2]$ of $9 \times 9$ dimension is simulated. A desired sharp synthetic edge $y_d[n_1, n_2]$ of identical dimension is also simulated. The output of the quadratic filter is $y[n_1, n_2] = \mathsf{X}^T[n_1, n_2]\mathsf{H_2}\mathsf{X}[n_1, n_2]$. Let the expected value of squared error between $y_d[n_1, n_2]$ and $y[n_1, n_2]$ is denoted as $\xi$.

$$\xi = E[|y_d[n_1, n_2] - \mathsf{X}^T[n_1, n_2]\mathsf{H_2}\mathsf{X}[n_1, n_2]|^2] \tag{5.1}$$

$\xi$ is minimized to yield an optimum $\mathsf{H}_2$, as plotted in Fig. 5.3.

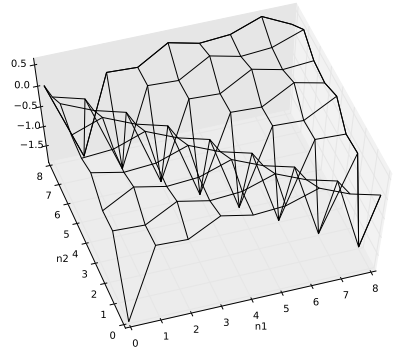



**Fig. 5.3.** Surface plot of the quadratic kernel

The kernel $\mathsf{H}_2$ is not completely isotropic. It becomes zero at the coordinates $(i, N-i)$ for $1 \leq i \leq N-1$, creating a wedge like minimum parallel to the main diagonal. There are local minima parallel to this. The $\mathsf{H}_2$ surface has high pass filter characteristics, making it suitable for edge detection. Once $\mathsf{H}_2$ is computed, it is required to implement it as a computationally simple structure. The direct implementation as in Eq. 2.13 results in large computational complexity. Instead, singular value decomposition[Gantmacher 1960] is performed on the kernel matrix $\mathsf{H}_2$ to yield an approximation $\tilde{\mathsf{H}_2}$ as

$$\tilde{\mathsf{H}_2} = \sum_{i=1}^{\rho} \lambda_i \mathsf{S_i} \mathsf{S_i}^T \tag{5.2}$$

The singular values and singular vectors of $\mathsf{H}_2$ are tabulated in Table 5.1. The parameter $\lambda_i$ are the singular values and each $\mathsf{S_i}$ is a $9 \times 1$ eigen vector. The value of $\rho$ is selected in such a manner that the Frobenius norm $||\mathsf{H}_2 - \tilde{\mathsf{H}_2}||$ is minimum. Each $\mathsf{S_i}$ is re-sized as a $3 \times 3$ FIR image filter that is equivalent to $\mathsf{H(i,j)}$ in Eq. 2.16. The outputs of FIR filters are squared and a weighted sum with $\lambda_i$ values yields the filter output. The structure of the filter is as in Fig. 5.4.

**Table 5.1:** Table of singular values and singular vectors of $H_2$

| $\lambda_i$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| 3.2783 | -0.3333 | -0.3333 | -0.3333 | -0.3333 |
| 2.9284 | -0.0570 | 0.4337 | -0.3768 | -0.0570 |
| 2.9284 | 0.4680 | -0.1846 | -0.2833 | 0.4680 |
| 2.8494 | -0.0388 | 0.2723 | 0.4559 | 0.4262 |
| 2.8494 | -0.4698 | -0.3848 | -0.1199 | 0.2013 |
| 2.8467 | 0.0998 | -0.2513 | 0.3726 | -0.4489 |
| 2.8467 | -0.4607 | 0.3988 | -0.2888 | 0.1440 |
| 2.7422 | 0.4690 | 0.0344 | -0.4570 | -0.1931 |
| 2.7422 | -0.0478 | -0.4702 | -0.1155 | 0.4300 |
| $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ |
| -0.3333 | -0.3333 | -0.3333 | -0.3333 | -0.3333 |
| 0.4337 | -0.3768 | -0.0570 | 0.4337 | -0.3768 |
| -0.1846 | -0.2833 | 0.4680 | -0.1846 | -0.2833 |
| 0.1971 | -0.1243 | -0.3875 | -0.4694 | -0.3317 |
| 0.4282 | 0.4547 | 0.2685 | -0.0434 | -0.3350 |
| 0.4711 | -0.4364 | 0.3491 | -0.2197 | 0.0638 |
| 0.0182 | -0.1783 | 0.3168 | -0.4171 | 0.4671 |
| 0.3900 | 0.3286 | -0.2759 | -0.4244 | 0.1285 |
| 0.2649 | -0.3381 | -0.3823 | 0.2053 | 0.4536 |

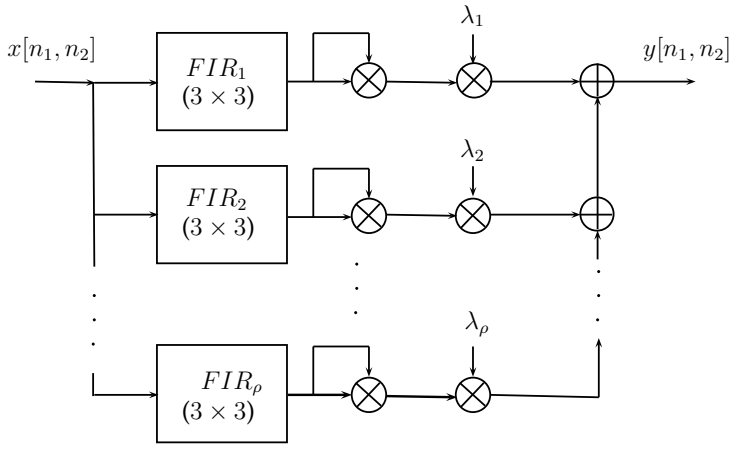



**Fig. 5.4.** Realization of $\mathsf{H}_2$ by singular value decomposition

## 5.5 Design of Experiment

The filter kernel $\tilde{\mathsf{H}_2}$ designed in the last section is simulated in Python with the help of scipy and pylab modules. In the first phase, $\tilde{\mathsf{H}_2}$ is tested for visual performance and quantitative parameters with the help of known image edges that are corrupted by impulsive and Gaussian noise of known variances. The filter is then included in the unsharp masking scheme and fingerprints are filtered. Two sets of fingerprints are used in the experiment. One set came from a fingerprint reader of $356 \times 328$ resolution. The second set is collected by a 12 megapixel digital camera. The prints are taken under noisy and poorly illuminated conditions. They are imported into Python using the image processing module and are subjected to unsharp masking. The filtered images are compared with those processed by unsharp masking based on edge detectors like Laplacian, Laplacian of Gaussian(LoG), Canny, Gabor and Sobel filters in terms of visual quality.

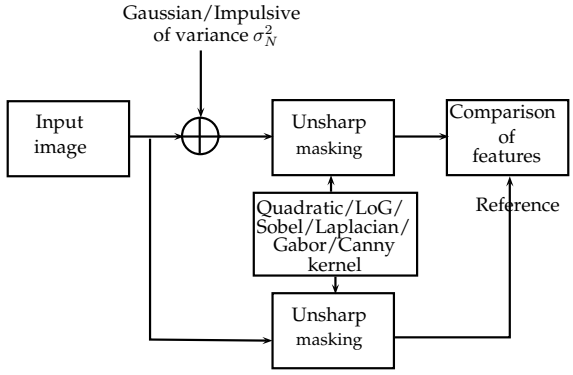



**Fig. 5.5.** Testing of unsharp masking

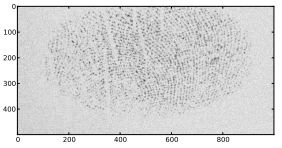
In the second phase of experiment, the performance of the unsharp masking scheme with quadratic filter is determined as in Fig. 5.5. Hundred fingerprints, each with dimension $356 \times 328$ pixels, corrupted by impulsive/Gaussian noise of known variance are applied to Sobel, LoG, Canny and quadratic filters and the outputs are compared. The improvement in signal to noise ratio is computed. The structural similarity index(SSIM) and sharpness of ridges are computed for all the hundred images for different additive noise variances and the average values of the parameters are computed. This is done to ensure the statistical soundness of the values of performance parameters.

In the final phase of experiment, the time of computation for various filters for input images of different dimensions are ascertained and compared. The results and analysis of the experiments are in Sec. 5.6.

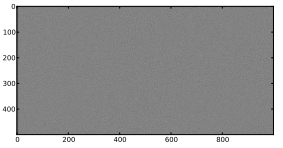
## 5.6 Results and Analysis

The filtering experiments conducted on noisy fingerprints with the quadratic filter yielded enhanced outputs that are visually better than

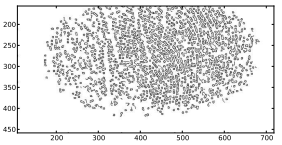
the ones yielded by unsharp masking with Laplacian, Sobel, Canny or LoG filters. Testing with impulsive noise gave rise to the results in Fig. 5.6. The input fingerprint corrupted by impulsive noise of variance 50 is shown in Fig. 5.6(a). Fig. 5.6(b) shows the result of unsharp masking based on Gabor filter. Although Gabor filter performs well in noiseless cases, it does not enhance noisy ridges. The scheme with Canny detector offered more enhancement in contrast but yielded broken ridges, especially in areas where the ridges are lost in noise as shown in Fig. 5.6(c). LoG filter yields continuous ridges but with limited enhancement as shown in Fig. 5.6(d). The gradient based Laplacian filter is very noise prone and the corresponding unsharp masking scheme yields no output as in Fig. 5.6(e). The output of unsharp masking using singular value decomposition based quadratic filter is as in Fig. 5.6(f). Quadratic filter gives out enhanced ridges even at an impulsive noise variance of 200, a noise level at which other filters do not distinguish the ridges. Fingerprints corrupted by Gaussian noise of variance 30 as in Fig. 5.7(a) is subjected to unsharp masking based on various filters. Unsharp masking using Gabor and Laplacian filters do not result in any enhancement as both the filters are susceptible to noise. LoG performs as shown in Fig. 5.7(b). Its abilty to enhance ridges is decreased with Gaussian noise even at lower variance than impulsive noise. The result of scheme based on Canny edge detector is shown in Fig. 5.7(c). Singular value decomposition based quadratic filter gives a visually enhanced output as in Fig. 5.7(d). It is imperative to quantify the visual quality of the outputs of unsharp masking based on various edge detection filters in terms of signal to noise ratio, sharpness of ridges, structural similarity index [Wang and Simoncelli 2004] and computational complexity. The first parameter is a measure of the noise invulnerability of the filter and the second parameter quantifies the crispness of the ridges. SSIM quantifies the degree of preservation of the overall structure of the fingerprint. The time of computation is indicative of the complexity of implementation.



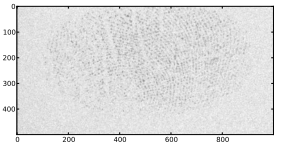
(a) Input fingerprint



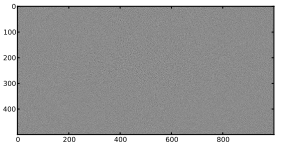
(b) Output with Gabor filter



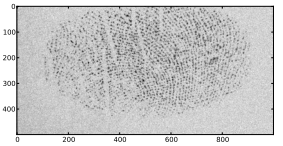
(c) Output with Canny filter



(d) Output with LoG filter

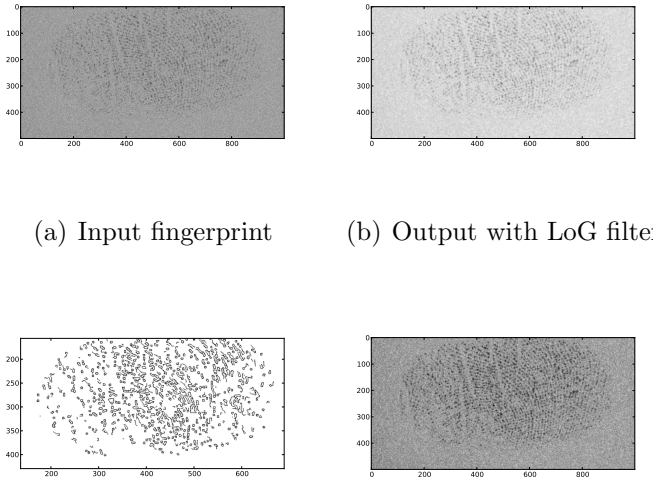


(e) Output with Laplacian filter



(f) Output with quadratic filter

**Fig. 5.6.** Outputs for impulsive noise variance 50



(a) Input fingerprint (b) Output with LoG filter

(c) Output with Canny filter (d) Output with quadratic filter

**Fig. 5.7.** Outputs of filters with Gaussian noise variance 30

### 5.6.1 Improvement in Signal to Noise Ratio and Peak Signal to Noise Ratio

The improvement in SNR and peak SNR are computed as per the experimental setup in Fig. 5.5. The SNR is expressed in dB as

$$SNR = 10\log_{10}\left[\frac{\sum_{n_1}\sum_{n_2} r^2_{[n_1,n_2]}}{\sum_{n_1}\sum_{n_2}[r^2_{[n_1,n_2]} - t^2_{[n_1,n_2]}]}\right] \tag{5.3}$$

The peak value of the SNR is expressed as

$$PSNR = 10\log_{10}\left[\frac{\max\left(r^2_{[n_1,n_2]}\right)}{\frac{1}{N_1N_2}\sum_{n_1}\sum_{n_2}[r^2_{[n_1,n_2]} - t^2_{[n_1,n_2]}]}\right] \tag{5.4}$$

where $r$ denotes the reference image and $t$ denotes the test image. $N_1N_2$ is the size of the image. The improvement in signal to noise ratio and in PSNR are tabulated in Table 5.2 when the input fingerprint is corrupted by impulsive noise. The values indicate that unsharp masking with quadratic filter performs best with impulsive noise.

**Table 5.2:** Improvement in SNR for various filters under impulsive noise

| Filter | SNR(dB) | PSNR(dB) |
|---|---|---|
| Quadratic | 20.13 | 21.89 |
| LoG | 16.16 | 18.59 |
| Canny | 13.45 | 15.36 |
| Laplacian | 10.77 | 11.20 |
| Sobel | 7.79 | 10.22 |

It has a $\approx 4dB$ advantage in signal to noise ratio compared to LoG filter. Table 5.3 shows the SNR improvement with Gaussian noise. The performance degrades for smaller noise variances but quadratic filter offers $\approx 2dB$ improvement in SNR compared with LoG filter.

### 5.6.2 Sharpness of Ridges

The sharpness of ridges in the enhanced fingerprint is decided by the noise invulnerability of the edge detection filter. A numerical figure of

**Table 5.3:** Improvement in SNR for various filters under Gaussian noise

| Filter | SNR(dB) | PSNR(dB) |
|---|---|---|
| Quadratic | 21.1 | 22.04 |
| LoG | 18.92 | 19.86 |
| Laplacian | 11.63 | 12.58 |
| Sobel | 10.56 | 13.50 |
| Canny | 9.33 | 12.10 |

merit of the edge detector[Thurnhofer and Mitra 1996] is given as

$$\kappa = \frac{1}{N_1 N_2} \sum_{n_1} \sum_{n_2} \frac{|\sigma^2_{l[n_1,n_2]_{test}} - \sigma^2_{l[n_1,n_2]_{ref}}|}{\sigma^2_{l[n_1,n_2]_{ref}} \mu_{l[n_1,n_2]_{ref}}} \tag{5.5}$$

$\sigma^2_{l[n_1,n_2]_{test}}$ is the localized variance (here a $3 \times 3$ pixel window is used to match the size of the filter mask) of the test image and $\sigma^2_{l[n_1,n_2]_{ref}}$ is that of the reference image. The localized mean of the reference image is $\mu_{l[n_1,n_2]_{ref}}$. For estimating the sharpness of ridges, hundred test images are formed by adding impulsive noise of variance ranging from 200 to 300 in steps of 5 to the fingerprint. These input images are then normalized and applied to unsharp masking based on Laplacian, LoG, Sobel, Canny and quadratic filters. The normalized, noisy input image is taken as the reference image and the normalized outputs of various filters are taken as test images. The variances and mean values are computed over a $3 \times 3$ pixel mask and summation is done all over the area of image. The hundred values each computed for 20 noise variance values are averaged and plotted as in Fig. 5.8. It

is seen that the parameter is the lowest for quadratic filter and it remains fairly constant as noise variance increases. This is indicative of the noise invulnerability and preservation of ridges of quadratic filter. LoG filter has a higher value and shows fluctuations as noise variance increases. Canny filter has a poorer value for $\kappa$. Laplacian and Gabor perform worse than Canny. The plots in Fig. 5.8 is consistent with the claim on visual quality based on Fig. 5.6. The entire procedure is

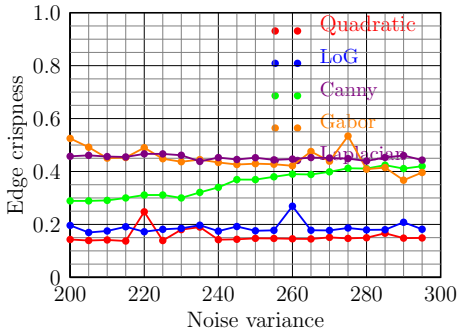



**Fig. 5.8.** Crispness of edges with impulsive noise

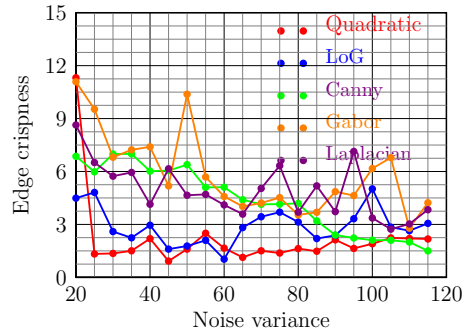



**Fig. 5.9.** Crispness of edges with Gaussian noise

repeated for Gaussian noise of variance ranging from 20 to 120 in steps of 5. The average sharpness of ridges for 100 fingerprints is plotted as in Fig. 5.9. The mean value of $\kappa$ is higher than that with impulsive noise, indicating the degradation of performance in presence of Gaussian noise. The parameter is the smallest for quadratic filters. LoG filter performs better than Laplacian, Canny and Gabor filters.

The parameter $\kappa$ should be zero when no noise is present and it increases monotonically as the degradation of edges by noise increases. Table 5.4 summarizes the values of the sharpness of ridges ($\kappa$) for the same levels of impulsive and Gaussian noise. The parameter is the lowest for unsharp masking with quadratic filter, indicating its noise invulnerability and the sharpness rendered to fingerprint ridges.

**Table 5.4:** Sharpness of ridges for various filters

| Filter | Impulsive $\kappa$ | Gaussian $\kappa$ |
|---|---|---|
| Quadratic | 0.1432 | 1.3547 |
| LoG | 0.1962 | 2.5951 |
| Sobel | 0.2990 | 3.1510 |
| Canny | 0.2908 | 6.9900 |
| Laplacian | 0.4574 | 5.7358 |

### 5.6.3 Structural Similarity Index(SSIM)

A common measure that is used to quantify the quality of an image is the mean square error[Eskicioglu and Fisher 1995], [Girod 1993], [Pappas and Safranek 2000],[Xu and Hauske 1994] which has many demerits. To overcome these demerits, Zhou Wang, et.al. proposed the structural similarity index (SSIM) to quantify the " visual quality"of the image[Wang and Simoncelli 2004],[Wang and Bovik 2002]. The similarity index between the images x and y is given as

$$SSIM(\mathsf{x},\mathsf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(2\mu_x^2 + 2\mu_y^2 + C_1)(2\sigma_x^2 + 2\sigma_y^2 + C_2)} \tag{5.6}$$

The parameters $\mu_x$ and $\mu_y$ are the means and $\sigma_x^2$ and $\sigma_y^2$ are the variances of x and y respectively. $\sigma_{xy}^2$ is the co-variance between x and y. $C_1$ and $C_2$ are non-zero constants included to avoid unstable results when $\sigma_x^2 + \sigma_y^2$ or $\mu_x^2 + \mu_y^2$ is very close to zero. When the images x and y are identical, the structural similarity index is unity and degrades

when the structural differences between x and y increases.

In the current experiment, structural similarity index is used to ascertain the improvement in the quality of fingerprints on unsharp masking using various edge detection filters. As shown in the experimental set up in Fig. 5.5, impulsive noise of variance ranging from 30 to 100 are added with hundred test fingerprints (x) and the unsharp masked outputs (y) using various edge detection filters are obtained. The SSIM between x and y are computed with $C_1 = C_2$ and plotted for different noise variance levels. The resulting plot is as shown in Fig. 5.10.

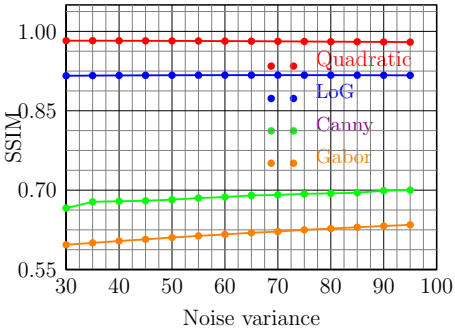



**Fig. 5.10.** SSIM for impulsive noise

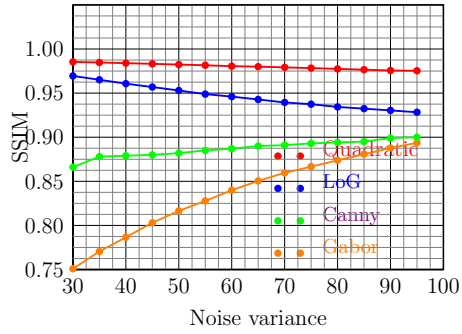



**Fig. 5.11.** SSIM for Gaussian noise

The expected values of SSIM are given in Table 5.5. Structural similarity index, indicative of the ability to enhance the fingerprints and the invulnerability to noise, is the largest for unsharp masking using quadratic filter. Its numerical value is approximately at 0.98. LoG filter has the next best value of SSIM with the numerical value approximately at 0.96. Also, SSIM for both quadratic and LoG remain constant as noise variance increases. Canny filter has an index approximately 0.3 below LoG and it increases as noise variance increases. SSIM for Gabor filter is still below that of Canny. The SSIM for

Laplacian is far smaller and is not shown on the graph.

**Table 5.5:** Sructural similarity index of images on unsharp masking using various edge detection filters for identical noise variances

| Filter | Mean SSIM (Impulsive) | Mean SSIM (Gaussian) |
|---|---|---|
| Quadratic | 0.9824 | 0.9854 |
| LoG | 0.9821 | 0.9695 |
| Canny | 0.6660 | 0.8660 |
| Gabor | 0.5968 | 0.7509 |
| Laplacian | 0.0018 | 0.0012 |

The computation of SSIM is repeated for hundred fingerprints corrupted by Gaussian noise of variance ranging from 30 to 100 using various filters and plotted as in Fig. 5.11. The graph shows a clear advantage in employing a quadratic filter. The SSIM ranges from 0.99 to 0.98 as Gaussian noise power varies from 30 to 100. Its mean SSIM is $\approx 0.16$ above that of LoG and it falls as the noise power increases. The SSIM for Canny filter is approximately 0.85 and is consistently above Laplacian and Gabor but below quadratic and LoG filters. Laplacian and Gabor filters perform equally bad in maintaining the structure of the image corrupted by Gaussian noise. The relatively high structural similarity index arising from quadratic filtering as given in Table 5.5 and the plots in Fig. 5.10 and Fig. 5.11 are consistent with the enhanced visual quality of latent fingerprints in Fig. 5.6 and Fig. 5.7.

## Computational Complexity

One major challenge in working with quadratic filter is the complexity arising from the computation of Kronecker products when a direct implementation is used. This difficulty is circumvented when approximate realization of the quadratic kernel, based on singular value decomposition, is employed. Now, only the principal components of $\mathsf{H}_2$ take part in the computation, resulting in significant reduction of complexity. Even with singular value decomposition, it is imperative to ascertain and compare the complexity of the filter with other edge detection filters. This comparison is accomplished by estimating the time taken for filtering in every case on identical images and identical computing platforms. A random $N \times N$ image is subjected to unsharp masking based on various filters discussed and the times of computation are found for different values of $N$ and are listed in Table 5.6. Canny filter is the slowest in extracting edges. It takes nearly 351 seconds to filter a $512 \times 512$ latent fingerprint image. The complexity increases linearly with the size of the image as indicated in Fig. 5.12.

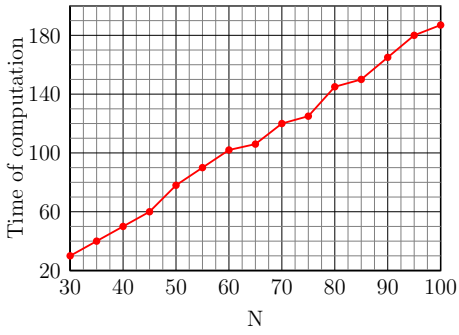



**Fig. 5.12.** Computational complexity for Canny filter

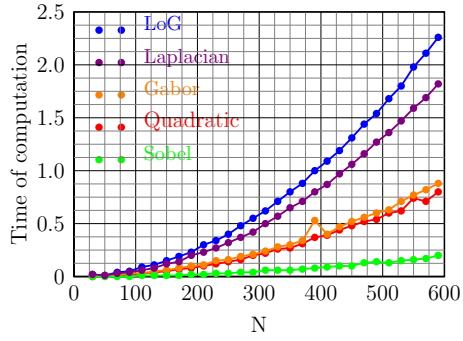



**Fig. 5.13.** Computational complexity for various filters

The time of computation for Laplacian, LoG, quadratic and Sobel are

as in Fig. 5.13. LoG filter is the second slowest, followed by Laplacian filter and Gabor filter respectively. Complexity of LoG and Laplacian follow almost a square law relationship with size of the image($N$) but that of Gabor filter increases at a smaller rate with $N$. It is observed that the time of computation for quadratic filter is consistently lower than LoG, Laplacian and Gabor filters. It takes 0.5932 seconds to filter a $512 \times 512$ latent fingerprint image. The main disadvantage of quadratic filters viz. the computational complexity is overcome by the singular value decomposition based implementation. It has roughly half the computations compared to Laplacian or LoG filters. Although Sobel filter is the fastest it failed to enhance the fingerprints.

**Table 5.6:** Time of computation in seconds for various image filters

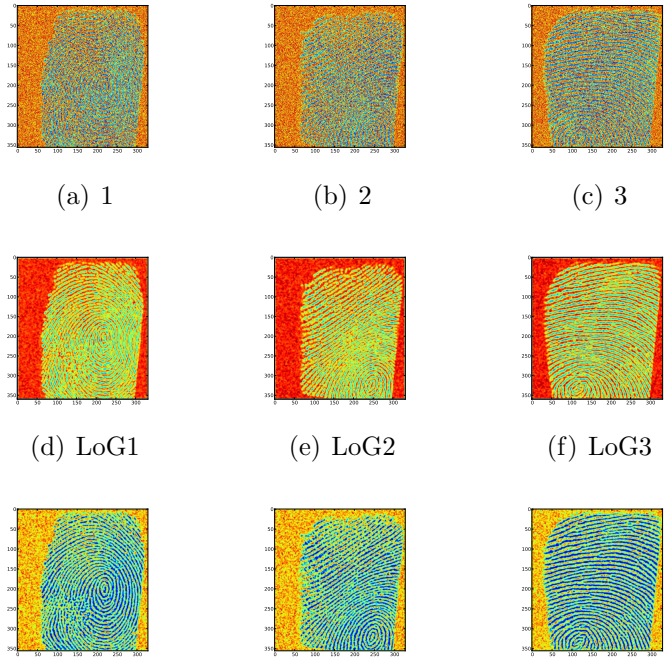
| Image size (Pixels) | Laplacian | LoG | Canny | Gabor | Quadratic (SVD) |
|---|---|---|---|---|---|
| $356 \times 328$ | 0.5615 | 0.70 | 289 | 0.2753 | 0.2555 |
| $512 \times 512$ | 1.368 | 1.688 | 351 | 0.6369 | 0.5932 |

Singular value decomposition based implementation of quadratic filter exhibits greater SNR and PSNR compared to other edge detectors and offers sharper fingerprint ridges. The largest structural similarity index between the uncorrupted fingerprint and the filtered version together with the small computation time required make quadratic filter a unique choice in the unsharp masking of latent fingerprints.

## 5.7 Inferences and Conclusion

A simple unsharp masking scheme employing a quadratic Volterra edge detection filter is used to enhance noisy fingerprints similar to

latent prints in crime scenes. The kernel of quadratic filter $\mathsf{H}_2$ is designed by Powell method of unconstrained optimization. As the direct implementation of $\mathsf{H}_2$ is computationally challenging, singular value decomposition is performed on $\mathsf{H}_2$ to yield a multichannel implementation that performs fast filtering. The salient feature of the unsharp masking is the noise invulnerability of edge detector. Most edge detectors based on spatial gradient perform poorly in presence of noise.



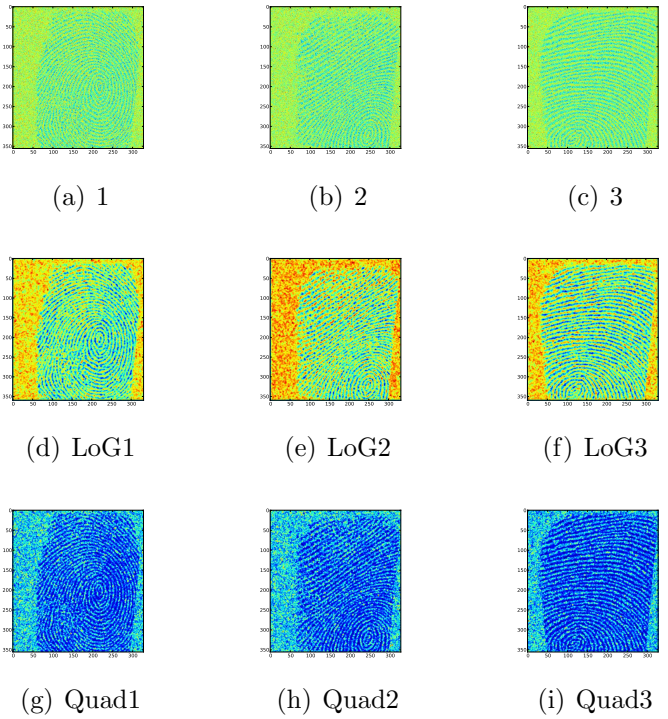
(a) 1 (b) 2 (c) 3

(d) LoG1 (e) LoG2 (f) LoG3

(g) Quad1 (h) Quad2 (i) Quad3

**Fig. 5.14.** Input fingerprints corrupted by impulsive noise(row 1) Outputs of LoG filter(row 2) Outputs of quadratic filter(row 3)

The quadratic filter is found to have better edge detection properties even under extremely noisy conditions. The average signal to noise ratio by unsharp masking with SVD based quadratic filter is approximately 2 to 4 dB above that with conventional filters with nearly half the time of processing. It also preserves the structural similarity and the sharpness of ridges better than conventional nonlinear filters.



(a) 1 (b) 2 (c) 3

(d) LoG1 (e) LoG2 (f) LoG3

(g) Quad1 (h) Quad2 (i) Quad3

**Fig. 5.15.** Input fingerprints corrupted by impulsive noise(row 1) Outputs of LoG filter(row 2) Outputs of quadratic filter(row 3)

A comparison between the outputs of unsharp masking, based on

quadratic and LoG filters, that are driven by fingerprints corrupted by impulsive noise of variance 200 is as shown in Fig. 5.14. The ridges are enhanced by the quadratic filter much better than that rendered by unsharp masking by LoG and other filters. Fig. 5.15 shows the comparison between the outputs of unsharp masking by quadratic and LoG filters driven by fingerprints corrupted by Gaussian noise of variance 150. The scheme based on quadratic filter has remarkable advantage in terms of enhancement compared to those based on LoG and other filters. It is thus concluded that the proposed quadratic filter based unsharp masking scheme is well suited for enhancing latent fingerprints from crime scenes and can outperform conventional filters.

Chapter 6

# Teager and Quadratic Filters for Retinal Images

## Contents

## 6.1 Introduction

Retina is the light sensitive tissue in the inner surface of the eye that serves as the screen to capture the image. Diabetic retinopathy is an eye disease as a result of vascular changes in the retina which, if left untreated, can lead to permanent loss of vision. This medical condition is caused by both Type 1 and Type 2 diabetes. The increased glucose and protein levels due to diabetes cause deposits and, consequently, swelling in the blood vessels in retina. Various stages in the progress of diabetic retinopathy fall into four groups as

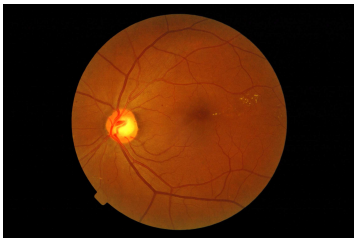
- Mild nonproliferative retinopathy
- Moderate nonproliferative retinopathy
- Severe nonproliferative retinopathy
- Proliferative retinopathy

**Mild Nonproliferative Retinopathy** It is a primitive stage in which small swellings called microaneurysms occur in the blood vessels in the retina. These appear as dots marked in Fig. 6.1(a).

**Moderate Nonproliferative Retinopathy** is a stage when some blood vessels are blocked. This is shown as in Fig. 6.1(b).

**Severe Nonproliferative Retinopathy** At this stage, more blood vessels are blocked, depriving of Oxygen in several areas of retina. These areas signal the body to grow more blood vessels in the retina. This results in greater swelling and further block in vessels. This stage of severe nonproliferative retinopathy is as shown in Fig. 6.1(c).

**Proliferative Retinopathy** is an advanced stage where the signals sent by the retina for nourishment initiate the growth of new blood vessels as indicated in Fig. 6.1(d). The abnormal blood vessels are fragile and can leak blood, causing permanent damage to the retina.



(a) Microaneurysms

(b) Moderate nonproliferative retinopathy

(c) Severe nonproliferative retinopathy

(d) Proliferative retinopathy

**Fig. 6.1.** Various stages in diabetic retinopathy

Diabetic retinopathy does not have any early warning signs or pain. As some symptoms are mistaken for other eye diseases, detection and early treatment are difficult, risking permanent damage to the retina

and loss of vision. Even the rapidly progressing macular edema, the condition in which the circular region at the center of the eye called macula is swelled is not easily detected as its main symptom viz. blurred vision is usually attributed to other eye ailments. Since the loss of vision due to diabetic retinopathy is permanent, it is imperative that early detection using fundoscopic methods and laser treatment should be done. And it becomes necessary to determine the periphery of microaneurysms exactly in fundoscopic images. The conventional Laplacian filtering or its modifications viz. Laplacian of Gaussian filtering yield blurred edges. Moreover, they are very susceptible to noise as noise appears as false edges in an image. Also it is observed that images contain polynomial nonlinearities, which are usually not accounted for in conventional edge detection filtering methods. So a polynomial based filtering method which can yield enhanced edges, while being invulnerable to noise, is needed. The work consists of the design and implementation of discrete Volterra series based quadratic filter for better edge detection, especially under noisy conditions.

## 6.2 Previous Work

Diabetic retinopathy, being a common diabetic complication leading to permanent blindness, there is research in digitizing and analyzing fundus images[Sleightholme et al. 1994]. Automatic detection of microaneurysms in fluorescein angiograms due to diabetic retinopathy has been reported [Spencer et al. 1991],[Øien 1995]. Screening and classification of diabetic retinopathy, based on Bayes method, has been done [Ege et al. 2000]. Neural network methods were applied in the automatic detection[Gardner and Elliot 1996]. Segmentation of retinal fundus images was proposed for the measurement of diabetic retinopathy lesions [Köse et al. 2011] [Mendonca and Nunes 1999].

The theory of Volterra functionals was developed by Vito Volterra to model nonlinear systems as parallel combinations of linear and polynomial systems of increasing order in the year 1887[Mathews and Sicuranza 2000]. Weiner applied Volterra series to Brownian motion to develop analytic functionals. Later the theory was applied to understand the effect of noise in a nonlinear radar receiver[Wiener 1942]. Volterra series was used in discrete system analysis by Alper[Alper 1963]. Recently, there is renewed interest in the design and implementation of polynomial systems, with the help of modern hardware and added computational resources. Unlike in linear system theory, there are no general methods to determine the Volterra kernels. Many methods were proposed for ascertaining the filter kernels,[Koh and Powers 1985],[Nowak and Veen 1994] especially the quadratic one.

Generation of images is by nonlinear processes and the human visual system is inherently nonlinear. These inherent nonlinearities in images are modeled with polynomial systems and many classes of quadratic filters were developed for edge preserving noise smoothing, edge extraction, texture determination, image interpolation etc. Quadratic filter was employed for edge detection [Ramponi 1986]. Teager filter was proposed as an energy detection operator[Kaiser 1990]. Edges are but spatial localization of energy in the image and Teager algorithm is employed in edge detection. Least squares method of designing class II filter, based on Teager algorithm, was proposed [Thurnhofer and Mitra 1996], the method that is relied on in Sec. 6.4.1.

## 6.3 Methodology

Type-II filters are proposed for minute edge detection that can detect the periphery of small microaneurysms due to diabetic retinopathy. Such precise detection can facilitate robotic surgery without loss of

unaffected cells. Three polynomial filters based on Volterra series are developed for this task. The work is broadly divided into three stages, *viz.*

- Design
- Implementation
- Testing

### 6.3.1 Design and Implementation

Two classes of Teager filters and a general quadratic filter are used for the precise detection of retinal microaneurysms. The flow of work is as shown in Fig. 6.2.

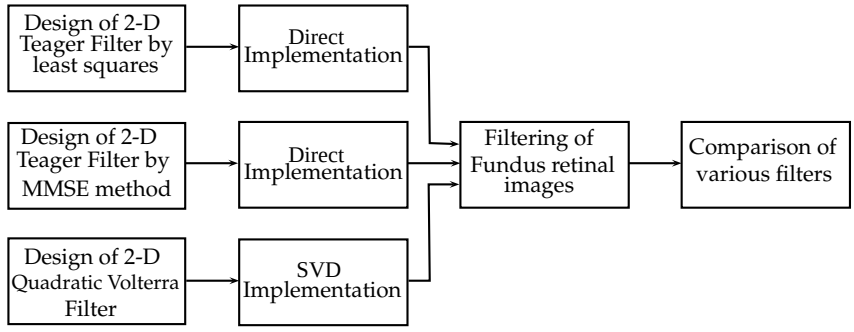



**Fig. 6.2.** Scheme of work

In the first stage of work, Teager filter based on least squares is designed and implemented, as elucidated in Sec. 6.4.1. The frequency response characteristics of this filter is computed and plotted. The frequency selectivity and edge detection features are also analyzed. The filter is used to detect the microaneurysms in retinal images. In

the second phase, the Teager filter is redesigned based on minimum mean square error criterion and the design validation steps are repeated. This phase is as detailed in Sec. 6.4.2. In the third phase, a general quadratic filter is designed based on Powell's optimization and is implemented using SVD method and applied on retinal images, as detailed in Sec. 6.4.3.

### 6.3.2 Testing

Once the three filters are found suitable for detecting microaneurysms of $\approx 25 - 100\mu m$ dimension[Spencer et al. 1991], the performance parameters such as noise invulnerability and the computational complexity are to be quantified.

#### Noise Performance

The performance of edge detection filters based on gradient method deteriorates as the level of additive noise increases. Teager and Quadratic filters are proposed as superior edge detectors with greater noise invulnerability. The claim is quantitatively validated by computing the signal to noise ratio, for images processed by quadratic and Teager filters and conventional filters like Laplacian, LoG, Sobel etc.[Pratt 2001]. The noise performance measures for the filters are summarized in Sec. 6.5.

#### Computational Complexity

The principal difficulty in working with quadratic filters is the enhanced complexity arising from the Kronecker products. Suitable matrix decompositions are applied to circumvent this difficulty. Analysis

of computational complexity in Teager, quadratic and conventional filters are carried out for validation of the design.

## 6.4 Design of 2-D Teager Filters for Edge Detection

Edges in images are formed by discontinuities in spatial, geometrical or photo-metric properties of objects[Pratt 2001]. Edge detection and extraction are image processing operations that find application in unsharp masking, remote sensing, object detection, machine vision, robotic surgery etc. Although edges are formed by high frequency components, simple high pass filtering does not suffice in detecting edges as it blurs the image. Generally, edges are detected by the computation of the derivative of the image. This computation is very noise sensitive since noise signals appear as false edges in an image. So the chief performance criterion of an edge detector becomes the invulnerability to noise. While conventional gradient based edge detectors such as Laplace filter, Sobel filter, Gaussian filter are very susceptible to noise, polynomial filters are observed to perform better in detecting edges even in presence of noise, both impulsive and Gaussian. Images are formed by nonlinear processes and human vision is inherently nonlinear. So, image processing and analysis by polynomial methods become a natural alternative. Most of the nonlinear components are modeled by the quadratic term alone.

Working of type-II Teager filters is based on the algorithm for detection of localizations of energy in a signal. Such filters are employed for detection of edges that manifest as localizations of energy. A good edge detector should maximize $(x[n_1+1, n_2+1]-x[n_1-1, n_2-1])^2$ for an arbitrary pixel $x[n_1, n_2]$. Two design strategies are presented for

Teager filters. First one is based on the method of least squares and the second method is the minimization of mean square error. These design steps, the realization of filters and their frequency characteristics are detailed in Sec. 6.4.1 and 6.4.2.

A 2-D Teager filter assumes the general model as

$$y[n_1, n_2] = x^2[n_1, n_2] - \sum_{k_1} \sum_{k_2} \phi_{k_1,k_2} x[n_1 - k_1, n_2 - k_2] x[n_1 + k_1, n_2 + k_2] \tag{6.1}$$

This filter, when excited by a two dimensional sinusoid $x[n_1, n_2] = \sin[\omega_0(an_1 + bn_2)]$, $-1 \leq a \leq 1$, $b = \sqrt{1 - a^2}$, yields a constant output for small $\omega_0$. The low frequency condition is easily met in images where possible frequencies are all well below $\frac{\pi}{2}$. The impact of $a$ on the two dimensional sinusoid are depicted in Fig. 6.3 to Fig. 6.6. The parameter $a$ controls the orientation of input image $x$. The higher the value of $a$, the more prominent is the effect of its negation. The filter should be designed in such a manner to work independent of the orientation of the input sinusoid.

Teager filter shows high pass characteristics and is employed for detecting edges in a noisy image. Design strategy, based on least square method, is proposed for detecting edges[Thurnhofer and Mitra 1996]. The frequency response of the system described by Eq.(6.1) is

$$H_2^{k_1,k_2}(\omega_1, \omega_2, \omega_3, \omega_4) = 1 - \sum_{k_1} \sum_{k_2} \phi_{k_1,k_2} \cos(k_1\omega_1 + k_2\omega_2 - k_1\omega_3 - k_2\omega_4) \tag{6.2}$$

The quadratic behaviour of the filter results in two frequency components corresponding to each spatial dimension and hence four indices in the frequency domain. The ac components in the output spectrum of the Volterra filter for a sinusoidal excitation $x[n_1, n_2] =$

$\sin[\omega_0(an_1 + \sqrt{1-a^2}n_2)]$ is

$$Y(\omega_1, \omega_2) = Y^{dc}(\omega_0, a)\delta(\omega_1, \omega_2) \tag{6.3}$$

where

$$\begin{aligned} Y^{dc}(\omega_0, a) = \pi^2[&H_2(\omega_0 a, \omega_0\sqrt{1-a^2}, -\omega_0 a, -\omega_0\sqrt{1-a^2}) \\ &+ H_2(-\omega_0 a, -\omega_0\sqrt{1-a^2}, \omega_0 a, \omega_0\sqrt{1-a^2})] \end{aligned} \tag{6.4}$$

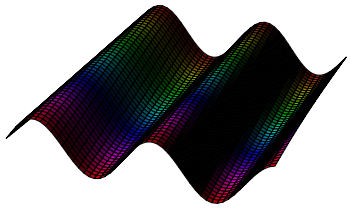


**Fig. 6.3.** 2-d sinusoid with $\omega = 0.45\pi$ and $a = 0.1$

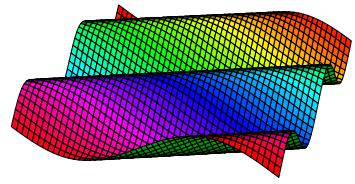


**Fig. 6.4.** 2-d sinusoid with $\omega = 0.45\pi$ and $a = 0.9$

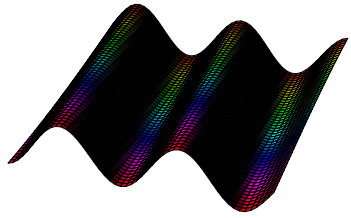


**Fig. 6.5.** 2-d sinusoid with $\omega = 0.45\pi$ and $a = -0.1$

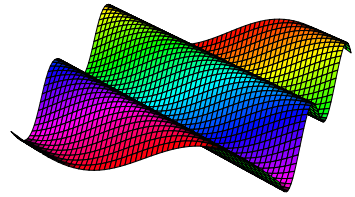


**Fig. 6.6.** 2-d sinusoid with $\omega = 0.45\pi$ and $a = -0.9$

Though the concept of frequency domain is questionable with polynomial systems, $Y^{dc}(\omega_0, a)$ is understood roughly as the frequency response of the filter as it is the frequency domain representation of the output when the system is driven by a two dimensional sinusoid. Its three dimensional representation for $\omega = [-\pi, +\pi]$ and for $a = [-1, +1]$ is representative of the frequency selective filtering characteristic. This feature is utilized in Sec. 6.4.1 and in Sec. 6.4.2.

### 6.4.1 Design of 2-D Teager Filter based on Least Squares Method

Method of least squares is used to design a filter of $5 \times 5$ mask, with $-2 \leq k_2 \leq 2$ and $0 \leq k_1 \leq 2$. These ranges provide 15 frequency responses of which $k_1 = k_2 = 0$, being irrelevant DC value, is eliminated. The frequency responses are pairwise identical for $k_1 = 0, k_2 = \pm 1$ and for $k_1 = 0, k_2 = \pm 2$ . Eq. 6.4 results in 12 frequency responses after eliminating the above three cases. The 12 functions are as tabulated in Table 6.1. These are sampled to obtain 12 vectors in the following manner. The span of $a$ is divided into 40 windows of width 0.05. The frequency range from 0 to $\frac{\pi}{2}$ is divided into 25 windows of dimension $\frac{\pi}{50}$. Each frequency window gives rise to 40 values of the frequency response for $-1 \leq a \leq 1$. The 25 frequency windows thus yield $25 \times 40$ values for each frequency response. The 12 such vectors each with 1000 values are arranged as a matrix $\mathsf{Y}$

$$\mathsf{Y} = [\,\mathsf{y}_1,\, \mathsf{y}_2,\, \cdots,\, \mathsf{y}_{12}\,] \tag{6.5}$$

The coefficients $\phi_{k_1,k_2}$, with $k_1$ and $k_2$ as indicated in Table 6.1, are arranged into another matrix $\phi$ as

$$\phi = [\,\phi_{0,1}\, \phi_{0,2}\, \cdots\, \phi_{2,2}\,] \tag{6.6}$$

The design problem is formulated as a matrix equation

$$\mathsf{d} = \mathsf{Y}\hat{\phi} + \mathsf{e} \tag{6.7}$$

Here $\mathsf{d}$ is the desired response and $\mathsf{e}$ is the error. $\hat{\phi}$ is the estimate of $\phi$ by the method of least squares. There is a well established solution for this equation[Mix 1995]. The said solution for Eq. (6.6) that minimizes $\mathsf{e}$ is

$$\hat{\phi} = [\mathsf{Y}^T\mathsf{Y}]^{-1}\mathsf{Y}^T\mathsf{d} \tag{6.8}$$

Here $\mathsf{d}$ is taken as a vector having 1000 samples of $\omega^2$. The scaled and approximated solutions $\phi_{k_1,k_2}$ are shown in the last column of Table 6.1.

**Table 6.1:** Frequency response functions and the filter coefficients

| Sl.No. | $k_1$ | $k_2$ | $Y^{dc}_{k_1,k_2}(\omega_0, a)$ | $\phi_{k_1,k_2}$ |
|---|---|---|---|---|
| 1 | 0 | 1 | $2\pi^2[1-\cos(2\omega_0 b)]$ | 0.0 |
| 2 | 0 | 2 | $2\pi^2[1-\cos(4\omega_0 b)]$ | -0.3 |
| 3 | 1 | -2 | $2\pi^2[1-\cos(2\omega_0 a-4\omega_0 b)]$ | 0.5 |
| 4 | 1 | -1 | $2\pi^2[1-\cos(2\omega_0 a-2\omega_0 b)]$ | 1.0 |
| 5 | 1 | 0 | $2\pi^2[1-\cos(2\omega_0 a)]$ | 0.5 |
| 6 | 1 | 1 | $2\pi^2[1-\cos(2\omega_0 a+2\omega_0 b)]$ | -1.0 |
| 7 | 1 | 2 | $2\pi^2[1-\cos(2\omega_0 a+4\omega_0 b)]$ | 0.0 |
| 8 | 2 | -2 | $2\pi^2[1-\cos(4\omega_0 a-4\omega_0 b)]$ | 0.5 |
| 9 | 2 | -1 | $2\pi^2[1-\cos(4\omega_0 a-2\omega_0 b)]$ | 0.3 |
| 10 | 2 | 0 | $2\pi^2[1-\cos(4\omega_0 a)]$ | 0.0 |
| 11 | 2 | 1 | $2\pi^2[1-\cos(4\omega_0 a+2\omega_0 b)]$ | 0.3 |
| 12 | 2 | 2 | $2\pi^2[1-\cos(4\omega_0 a+4\omega_0 b)]$ | 0.0 |

The relation between $\phi_{k_1,k_2}$ and the coefficients of kernel matrix is

$$h_2[k_1,k_2,-k_2,-k_2] = \begin{cases} -\phi_{k_1,k_2}; & \substack{n_1+|n_2|\neq 0 \\ 0\leq n_1\leq\infty, \\ -\infty\leq n_2\leq\infty} \\ 0; & n_1 \leq 0, \end{cases} \tag{6.9}$$

The filter is realized as

$$\begin{aligned} y[n_1,n_2] = {} & 1.8x^2[n_1,n_2] - x[n_1-1,n_2+1]x[n_1+1,n_2-1] \\ & + x[n_1-1,n_2-1]x[n_1+1,n_2+1] \\ & - 0.5x[n_1-1,n_2+2]x[n_1+1,n_2-2] \\ & - 0.3x[n_1-2,n_2+1]x[n_1+2,n_2-1] \\ & - 0.3x[n_1-2,n_2-1]x[n_1+2,n_2+1] \\ & + 0.3x[n_1,n_2-2]x[n_1,n_2+2] \\ & - 0.5x[n_1-1,n_2]x[n_1+1,n_2] \\ & - 0.5x[n_1-2,n_2+2]x[n_1+2,n_2-2] \end{aligned} \tag{6.10}$$

The isotropy of this filter is understood from the frequency response function $\Upsilon^{dc}(\omega_0,a)$. Substituting the filter coefficients in Eq. (6.4)

$$\begin{aligned} \Upsilon^{dc}(\omega_0,a) = {} & 2\pi^2[1.8 - \cos(2\omega_0\sqrt{1-a^2} - 2\omega_0 a) \\ & - 0.5\cos(4\omega_0\sqrt{1-a^2} - 2\omega_0 a) \\ & - 0.3\cos(2\omega_0\sqrt{1-a^2} - 4\omega_0 a) \\ & - 0.3\cos(2\omega_0\sqrt{(1-a^2)} + 4\omega_0 a) \\ & - 0.5\cos(\omega_0\sqrt{1-a^2} - \omega_0 a)] \end{aligned} \tag{6.11}$$

This frequency response function is plotted in Fig. 6.7. It is seen that for $\omega_0 = 0$ the filter output is zero irrespective of $a$, the orientation of the input image. As $\omega_0$ increases from 0 onwards, the response increases and reaches the maximum value at $\omega_0 = \frac{\pi}{4}$. Then it starts

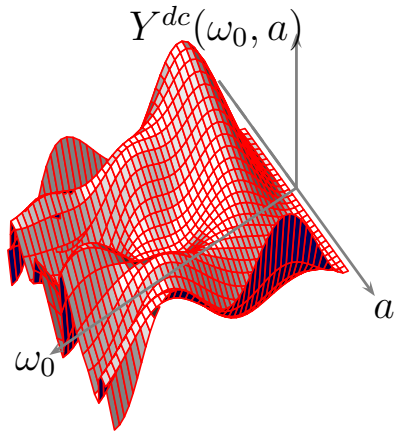



**Fig. 6.7.** Isotropy plot of Teager filter(LS)

decreasing, indicating high pass characteristics. The filter has the desirable property of being isotropic up to $\omega_0 = \frac{\pi}{4}$. From $\omega_0 = \frac{\pi}{4}$ onwards the behaviour is not entirely isotropic. The roll off is not uniform but the response falls quickly for $a = -0.25$.

### 6.4.2 Design of Teager Filter based on Minimum Mean Square Error

Alternate design strategy is sought to improve the high pass characteristics of 2-D Teager filter by resorting to minimum mean square error method. Hence, the coefficients for the Teager filter $\phi_{k_1,k_2}$ in Eq. (6.1) are to be designed based on the minimization of mean square error between a desired image and the actual image. In this work, a blunt synthetic edge $\mathsf{x}[n_1, n_2]$ of $50 \times 50$ dimension and a desired sharp edge $\mathsf{y_d}[n_1, n_2]$ with the same dimension are simulated. The mean square

error between $\mathsf{y_d}[n_1, n_2]$ and $\mathsf{y}[n_1, n_2]$ is

$$\zeta = E[|\mathsf{y_d}[n_1, n_2] - \mathsf{y}c[n_1, n_2]|^2] \tag{6.12}$$

Mean square error is minimized using Powell method[Fletcher and Powell 1963] to yield the coefficients as in Table 6.2. This unconstrained optimization is preferred as it offers fast convergence. The filter is realized as

$$\begin{aligned} y[n_1, n_2] = x^2[n_1, n_2] &- 0.5x[n_1, n_2 - 2]x[n_1, n_2 + 2] \\ &+ 0.5x[n_1 - 1, n_2 - 2]x[n_1 + 1, n_2 + 2] \\ &+ x[n_1 - 1, n_2 + 1]x[n_1 + 1, n_2 - 1] \\ &- x[n_1 - 1, n_2 - 1]x[n_1 + 1, n_2 + 1] \\ &- 0.25x[n_1 - 2, n_2 + 2]x[n_1 + 2, n_2 - 2] \\ &- 0.25x[n_1 - 2, n_2 + 1]x[n_1 + 2, n_2 - 1] \\ &- 0.5x[n_1 - 2, n_2]x[n_1 + 2, n_2] \\ &- 0.5x[n_1 - 2, n_2 - 2]x[n_1 + 2, n_2 + 2] \end{aligned} \tag{6.13}$$

The frequency response function is

$$\begin{aligned} Y^{dc}_{(\omega_0, a)} = 2\pi^2[1 &- 0.5\cos(4\omega_0\sqrt{1-a^2}) \\ &+ 0.5\cos(4\omega_0\sqrt{1-a^2} + 2a) \\ &+ \cos(2\omega_0 a - 2\omega_0\sqrt{1-a^2}) \\ &- \cos(2\omega_0 a + 2\omega_0\sqrt{1-a^2}) \\ &- 0.25\cos(4\omega_0 a - 4\omega_0\sqrt{1-a^2}) \\ &- 0.25\cos(4\omega_0 a - 2\omega_0\sqrt{1-y^2}) \\ &- 0.5\cos(4\omega_0 a) \\ &- 0.5\cos(4\omega_0 a + 4\omega_0\sqrt{1-a^2})] \end{aligned} \tag{6.14}$$

The function is viewed from the top as in Fig. 6.8.

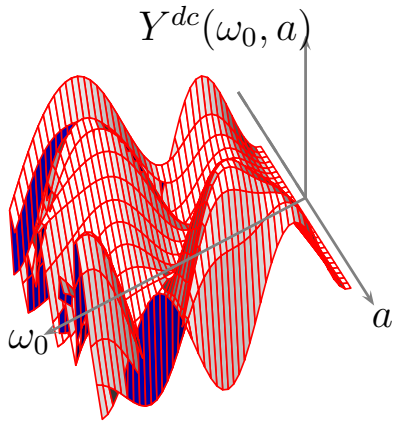



**Fig. 6.8.** Isotropy plot of Teager filter(MMSE)

**Table 6.2:** Filter coefficients by minimization of mean square error

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $k_1$ | 0 | 0 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| $k_2$ | 0 | -2 | -1 | 2 | 1 | -2 | -1 | 0 | 2 |
| $\phi_{k_1,k_2}$ | 1 | -0.5 | 1 | 0.5 | -1 | -0.25 | -0.25 | -0.5 | -0.5 |

The response is not as isotropic as the filter governed by Eq. 6.10. But the filter has better high pass characteristics especially for small values of a. The response is zero up to $\omega_0 = \frac{\pi}{2}$ and then it gradually rises, passing the high frequency components and then falls to zero at $\omega_0 = \pi$.

### 6.4.3 Design of Quadratic Edge Filter

The Teager filters in the last sections yield sharp edges, making the localization of exudates easier than that with linear filters. However

greater enhancement of edges is possible with a general quadratic filter governed by the Eq. 2.13. This section presents the realization of an isotropic quadratic filter based on optimization methods. A blunt input synthetic edge $\mathsf{X}[n_1, n_2]$ and a desired sharp synthetic edge $\mathsf{Y_d}[n_1, n_2]$ each of $9 \times 9$ dimension are simulated. The output of the quadratic filter is $\mathsf{Y}[n_1, n_2] = \mathsf{X}^T[n_1, n_2]\mathsf{H_2}\mathsf{X}[n_1, n_2]$. Let the mean square error between $\mathsf{Y_d}[n_1, n_2]$ and $\mathsf{y}[n_1, n_2]$ be $\xi$.

$$\xi = E[|\mathsf{y_d}[n_1, n_2] - \mathsf{X}^T[n_1, n_2]\mathsf{H_2}\mathsf{X}[n_1, n_2]|^2] \tag{6.15}$$

$\xi$ is minimized using Powell method to yield an optimum $\mathsf{H_2}$ and is plotted as in Fig. 6.9. The kernel is both symmetric and isotropic.

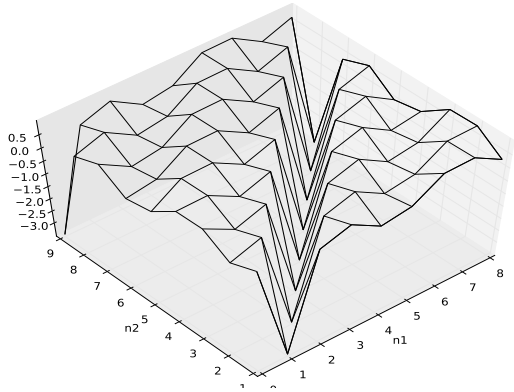



**Fig. 6.9.** Surface plot of the quadratic kernel

A direct implementation in the form of Eq. 2.13 is not feasible as indicated in Fig. 4.4 in page (56). Instead SVD decomposition [Gantmacher 1960] is performed on $\mathsf{H_2}$ to yield an approximation $\tilde{\mathsf{H}}_2$ as

$$\tilde{\mathsf{H}}_2 = \sum_{i=1}^{\rho} \lambda_i \mathsf{S_i}\mathsf{S_i}^T \tag{6.16}$$

where $\lambda_i$ are the eigen values and each $\mathsf{S_i}$ is a $9 \times 1$ eigen vector. The singular values and singular vectors of $\mathsf{H_2}$ are tabulated in Table 6.3.

$\rho$ is selected in such a manner that the Frobenius norm $||\mathsf{H}_2 - \tilde{\mathsf{H}_2}||$ is minimum. Each $\mathsf{S_i}$ can be re-sized as a $3 \times 3$ FIR image filter that is equivalent to $H(i,j)$ in Eq. 2.16. The filter structure is as in Fig. 5.4 in page 77. The outputs of FIR filters are squared and a weighted sum with $\lambda_i$ values yields the filter output.

## 6.5 Results

Teager filters based on least squares method and minimum mean square error criterion are designed and tested with retinal images. The filters are capable of enhancing microaneurysms typically of $100\mu m$ size, making the localization easy for laser surgery. The images processed by various edge detection filters like Laplacian, LoG, Sobel are compared with Teager filters in Fig. 6.10. Obviously, Sobel filter shows the poorest performance in localizing the microaneurysms. The edges appear blurred and irregularities in the background are also magnified, making detection very difficult. Laplacian gives a better outline of microaneurysms as in Fig. 6.10(b). Here microaneyurisms are contrasted from the background but the edges are still not sharp enough. Laplacian of Gaussian (LoG) filter has better edge detection feature than Laplacian as in Fig. 6.10(d) and has better noise performance. Teager filters are observed to perform better than linear filters. Fig. 6.10(e) shows the output of Teager filter by least squares given by Eq. 6.10. The edge detection characteristics of Teager filter by the method of least squares is superior to that of LoG filter. The edges are easily contrasted from the background. Fig. 6.7 shows that the high pass characteristics are slightly dependent on the orientation of the input image. The Teager filter by MMSE criterion contrasts the edges of microaneurysms from the background. Fig. 6.8 shows that, though orientation dependent, filter by MMSE has a better high pass characteristics from $\omega = \frac{\pi}{2}$ to $\omega = \pi$. Fig. 6.10(f) shows the output of this

**Table 6.3:** Table of singular values and singular vectors of $H_2$

| $\lambda_i$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| 4.2698 | 0.4714 | 0.0000 | -0.4714 | 0.0000 |
| 4.2698 | -0.2357 | 0.4082 | -0.3611 | 0.3030 |
| 4.1421 | -0.2357 | -0.4082 | -0.0819 | 0.4642 |
| 4.1421 | 0.4714 | 0.0000 | 0.2357 | 0.4083 |
| 4.0016 | -0.2357 | 0.4082 | 0.4430 | 0.1612 |
| 4.0016 | -0.2357 | -0.4082 | 0.4430 | -0.1612 |
| 3.9921 | 0.4714 | 0.0000 | 0.2357 | -0.4083 |
| 3.9921 | -0.2357 | 0.4082 | -0.0819 | -0.4642 |
| 0.7054 | -0.2357 | -0.4082 | -0.3611 | -0.3030 |

| $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ |
|---|---|---|---|---|
| 0.0000 | -0.4714 | 0.0000 | 0.4714 | -0.3333 |
| 0.1612 | 0.4430 | 0.4642 | 0.0819 | -0.3333 |
| -0.3030 | -0.3611 | 0.1612 | -0.4430 | -0.3333 |
| 0.4083 | 0.2357 | -0.4083 | -0.2357 | -0.3333 |
| -0.4642 | -0.0819 | -0.3030 | 0.3611 | -0.3333 |
| 0.4642 | -0.0818 | 0.3030 | 0.3611 | -0.3333 |
| -0.4083 | 0.2357 | 0.4083 | -0.2357 | -0.3333 |
| 0.3030 | -0.3611 | -0.1612 | -0.4430 | -0.3333 |
| -0.1612 | 0.4430 | -0.4642 | 0.0819 | -0.3333 |

filter. Here not only the edges are sharper and easy localization is possible, but the contrast with the background is greater than provided by Teager filter by least squares.

The general quadratic filter, designed by optimization and implemented by singular value decomposition as in Fig. 4.8, extracts edges better than Teager filters. The image processed by the quadratic filter is as in Fig. 6.11. The filter detects edges irrespective of the orientation of input image, since its kernel is symmetric and isotopic, as indicated in Fig. 6.9. This feature results in the enhancement of the periphery of small microaneurysms in the output image.

### 6.5.1 Noise Performance of Edge Filters

Additive noise introduces false edges in images and most conventional filters respond to these false edges as well. While conventional edge detection filters like Laplacian, LoG filters cannot distinguish these false edges, quadratic filter is very insensitive to false edges due to noise. The noise performance of quadratic and Teager filters is tested by the experimental set up in Fig. 6.14.

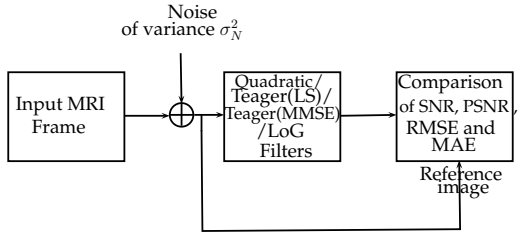
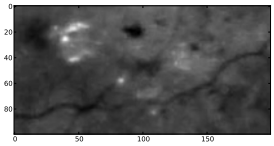
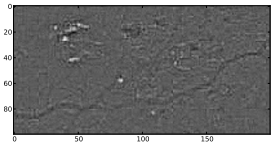



**Fig. 6.14.** Experimental setup for measuring signal to noise ratio

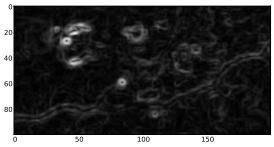
The filters developed are tested with retinal images corrupted by Gaus-



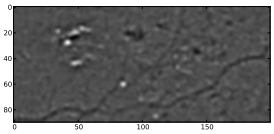
(a) microaneurysms in the input image



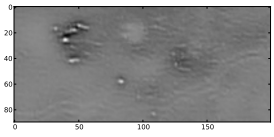
(b) Laplacian filter



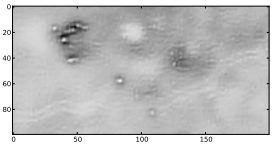
(c) Sobel filter



(d) LoG filter

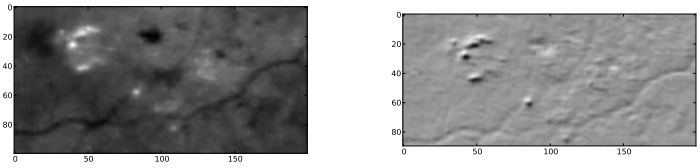


(e) Teager filter(LS)



(f) Teager filter(MMSE)

**Fig. 6.10.** Outputs of Teager filters, Laplacian, LoG and Sobel filters



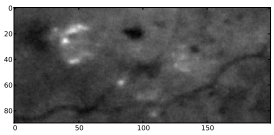
(a) microaneurysms in the input image (b) Output of quadratic filter

**Fig. 6.11.** Input microaneurysms and enhancement by quadratic filter

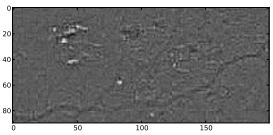
sian and impulsive noises. Gaussian noise of variances 5 and 20 are added with the image of microaneurysms and are passed through various filters. The results are in Fig. 6.12 and Fig. 6.13.

Laplacian filter is very noise prone while LoG is more invulnerable. Teager filter by minimization of mean square error is more robust than the one by least squares. Quadratic filter undoubtedly has the best invulnerability towards impulsive noise. Its outptut is noise insensitive even when the Gaussian noise variance $\sigma_N^2$ becomes 20 and both Teager filters yield noisy outputs.

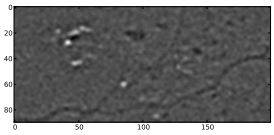
Noise in image processing is rarely Gaussian but impulsive in nature. The filters are tested with retinal images corrupted by impulsive noise. Fig. 6.15 shows the visual output of various filters. Laplacian and LoG yield noisy outputs while the Teager and quadratic perform better. Teager filter by MMSE gives out sharper edges than the one by least squares. Quadratic filter also gives out noise free edges.



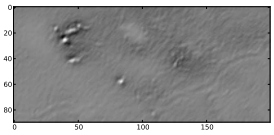
(a) Input image with Gaussian noise



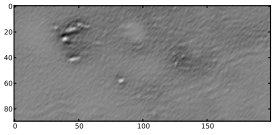
(b) Laplacian filter



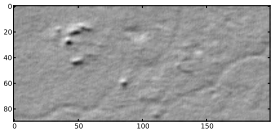
(c) LoG filter



(d) Teager filter(LS)

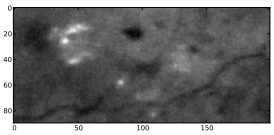


(e) Teager filter(MMSE)

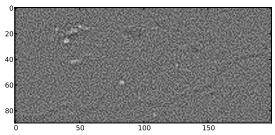


(f) quadratic filter

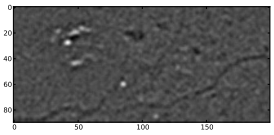
**Fig. 6.12.** Various filter outputs for images corrupted by Gaussian noise of variance $\sigma_N^2 = 5$



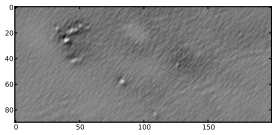
(a) Input image with Gaussian noise



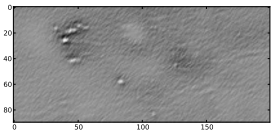
(b) Laplacian filter



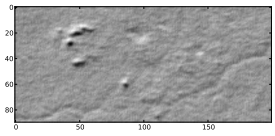
(c) LoG filter



(d) Teager filter(LS)



(e) Teager filter(MMSE)



(f) quadratic filter

**Fig. 6.13.** Various filter outputs for images corrupted by Gaussian noise of variance $\sigma_N^2 = 20$

The performance of the filters in presence of noise can be quantitatively understood by adding impulsive and Gaussian noise of variance 20 to the fundus image, passing them through filters and comparing the outputs with the outputs for noiseless image inputs. The various quality factors like signal to noise ratio(SNR), peak signal to noise ratio(PSNR), root mean square error(RMSE), mean absolute error(MAE) are listed in Table 6.4.

**Table 6.4:** Quality factors of various filters in presence of noise

| Noise | Filter | SNR (dB) | PSNR (dB) | RMSE | MAE |
|---|---|---|---|---|---|
| Gaussian | Laplacian | 20.49 | 24.89 | 14.52 | 8.23 |
| | LoG | 25.23 | 29.66 | 8.38 | 4.64 |
| | Teager(LS) | 30.16 | 34.65 | 4.72 | 2.35 |
| | Teager(MMSE) | 30.65 | 35.14 | 4.46 | 1.93 |
| | Quadratic | 25.20 | 29.94 | 8.11 | 4.42 |
| Impulsive | Laplacian | 26.14 | 30.53 | 7.58 | 4.26 |
| | LoG | 34.20 | 38.63 | 2.99 | 1.64 |
| | Teager(LS) | 36.15 | 40.84 | 2.31 | 1.35 |
| | Teager(MMSE) | 32.30 | 36.79 | 3.69 | 2.34 |
| | Quadratic | 35.97 | 40.71 | 2.35 | 1.27 |

With Gaussian noise, the SNR and the PSNR of quadratic and LoG filter are comparable and are $\approx 5dB$ higher than those of Laplacian filter. Teager filters have SNR and PSNR $\approx 5dB$ above those of quadratic and LoG filters. With impulsive noise, Teager(LS) and

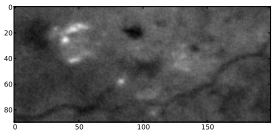
quadratic filters have similar SNR and PSNR values and are $\approx 2dB$ above those of LoG filter.

## 6.6 Discussion

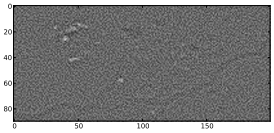
Quadratic edge filters are employed for the enhancement and localization of microaneurysms caused by diabetic retinopathy. Teager filters by least squares and by the minimization of mean square error are implemented. They are observed to have better edge detection capability as well as noise invulnerability than Laplacian, Sobel and LoG filters. A quadratic filter with symmetric and isotropic kernel is designed based on Powel method of optimization. This filter is implemented based on singular value decomposition of the kernel. Besides the performance in presence of noise, the computation times of various filters are of interest. They are computed on an intel core $i3$ processor and are tabulated in Table 6.5 for two images of different sizes.

**Table 6.5:** Time of computation in seconds for various filters

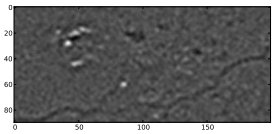
| Filter | Image size $(356 \times 328)$ | Image size $(504 \times 671)$ |
|---|---|---|
| Sobel | 0.04 | 0.11 |
| Laplacian | 0.58 | 1.70 |
| LoG | 0.64 | 1.86 |
| Teager(LS) | 2.75 | 7.96 |
| Teager(MMSE) | 2.38 | 6.97 |
| Quadratic(SVD) | 0.64 | 1.86 |



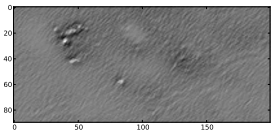
(a) Input image with gaussian noise



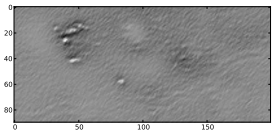
(b) Output of Laplacian filter



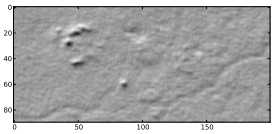
(c) Output of LoG filter



(d) Output of Teager filter(LS) filter



(e) Output of Teager filter(MMSE)



(f) Output of quadratic filter

**Fig. 6.15.** Various filter outputs for images corrupted by impulsive noise of variance $\sigma_N^2 = 20$

The SVD implementation of quadratic filter is faster than Teager filters while maintaining a similar noise performance. The Teager filters are the slowest, although they have good edge extraction characteristics and noise performance. It is concluded that Teager and quadratic filters outperform conventional edge detection filters in the enhancement and localization of retinal microaneurysms. Among the two, quadratic filter is faster in execution than Teager filters.

# Part III

# Quadratic Filters for Noise Removal

# Foreword to Part III

Magnetic Resonance Imaging(MRI) is one of the key advancements in the electronic aided medical diagnosis. Several eminent scientists contributed to the development of the MRI machine of which many were awarded the Nobel prize for MRI related contributions. In this radical method, nuclear resonances in body tissue due to applied magnetic field are mapped into light intensity variations, forming an image. High resolution is achieved by rapidly changing the heavy magnetic field applied. Such unsteady magnetic field makes the acquisition of the image very noisy. Impulsive noise, in particular, gets added with the images generated. Unlike Gaussian noise that is common in communication systems, it is difficult to get rid of impulsive noise. Conventional nonlinear filters employed for noise removal from images are median filter, mean filter, Gaussian filter etc. Such filters often suffer from poor edge resolution, blurring and poor signal to noise ratio. As an alternative, polynomial filters based on Volterra series is proposed for preprocessing raw MRI data. Part III of the thesis deals with the design and implementation of a two dimensional quadratic filter. It is verified that the new filter offers $\approx$ 10 dB improvement in SNR compared to conventional spatial filters. The noise removal filter is succeeded by an unsharp masking scheme employing a quadratic edge detection filter. This step is done for accomplishing further improvement in contrast. The design strategy, implementation and performance parameters for the quadratic noise removal filter are summarized in chapter 7.

Chapter 7

# Spatial Filtering of MRI Images for the Removal of Impulsive Noise using a Quadratic Volterra Filter

## Contents

## 7.1 Introduction

Nuclear magnetic resonance imaging(MRI) is a popular radio-graphic imaging method to view the internal structure of the human body. This imaging technique is very vital in the early detection of tumours and other malignant deposits. In this imaging technique, the nuclear resonances with applied magnetic field are transformed into light intensity variations, forming an image of the body part under study. The magnetic resonance is based on Larmor equation which states that the frequency of precession of magnetic moment ($\omega$) is directly proportional to the magnetic field strength ($B$) and the gyro-magnetic ratio ($\gamma$). It is at this frequency that the nucleus absorbs energy. Due to this absorption, the proton alters its alignment. Methods were devised for measuring this change in magnetic moment [Rabi and Kusch 1938]. Later Bloch and Purcell discovered that when certain nuclei were placed in a magnetic field they absorbed energy in the electromagnetic spectrum,[Bloch and Packard 1946],[Purcell and Pound 1946] and re-emitted this energy when the nuclei returned to their original state. The strength of the magnetic field and the radio frequency matched each other according to the Larmor relationship[Damadian and Field 1976a], [Damadian and Field 1976b]. The relaxation times in normal and cancerous tissue were observed to be different. In 1974, Peter Mansfield and Paul C Lauterbur used these magnetic field gradients to get the spatial location of NMR signals. In 1975, Fourier transform of phase and frequency encoding was employed for reconstructing two dimensional images and this formed the origin of the present day MRI. An excellent history of the contributions in nuclear magnetic resonance imaging is presented in the paper on cardiac MRI[Geva 2006].

In the present day scenario, MRI is done under a heavy magnetic field, typically 1.5 Tesla to 2 Tesla. The large amount of electromagnetic interference adds impulsive noise with the MRI images. Often, it is dif-

ficult to remove impulsive noise from image signals. The conventional image filters do not take into account the mechanism of generation of the image, which in the case of most natural images is nonlinear. The logical solution is to employ a nonlinear filter to account for the nonlinear effects in image generation. Polynomial nonlinearities can be modeled by truncated Volterra series that adds quadratic, cubic and higher order systems in parallel with a linear system. It has been observed that much of the nonlinear effects can be modeled by the quadratic filter alone. The task under consideration is to design and implement a quadratic Volterra filter that can account the inherent nonlinearities in image formation while removing the impulsive noise.

## 7.2 Methodology

The methodology of noise removal is as outlined in Fig. 7.1. The first phase is the design of the quadratic kernel for noise removal $\mathsf{H}_{2_{\text{noise}}}$ by Powell method of optimization as detailed in Sec. 7.3.

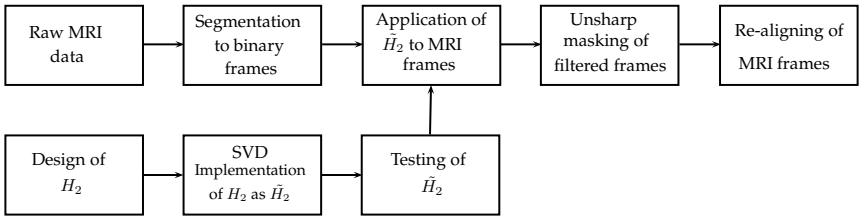



**Fig. 7.1.** Flow of work

This step of optimization minimizes the mean square error between a known image and the output of a quadratic filter that receives the noisy version of the image at its input. The optimization is done repeatedly until minimum mean square error yields. The solution for $\mathsf{H}_2$ that attains this minimum error is selected as the filter kernel. Since the direct implementation of this kernel is computationally complex,

singular value decomposition is done on $\mathsf{H}_{2_{\text{noise}}}$ to yield an approximate realization $\tilde{\mathsf{H}}_{2_{\text{noise}}}$ as in Fig. 4.8. In the third phase, the kernel $\tilde{\mathsf{H}}_{2_{\text{noise}}}$ is tested with known images for ascertaining the improvement in signal to noise ratio and peak signal to noise ratio before applying to noisy MRI images. These are as detailed in Sec. 7.6. The raw MRI data is taken and segmented to individual frames. The noisy frames are subjected to filtering by $\tilde{\mathsf{H}}_{2_{\text{noise}}}$ as implemented in Fig. 4.8 in page 64 for removing impulsive noise. In the last phase of work, unsharp masking is done to enhance the contrast of the filtered frames. In this step, a scaled version of the frames are added with the frames themselves to improve the high frequency components. The work relies on a quadratic edge detection filter, designed as in Sec. 7.5, rather than conventional edge detectors like Canny, Prewitt or Laplacian of Gaussian since the former provides better edge crispness and invulnerability to noise. This filter is also implemented based on SVD decomposition as in Fig. 4.8 in page 64. The analysis of results is done and is summarized in Sec. 7.7.

## 7.3 Design and Implementation of Noise Removal Filter

It is proposed that a quadratic filter can effect greater improvement in signal to noise ratio compared to conventional mean and median filters. The principal issue in employing a quadratic filter is the identification of its kernel $\mathsf{H}_{2_{\text{noise}}}$. Powell optimization is used for obtaining $\mathsf{H}_{2_{\text{noise}}}$ [Fletcher and Powell 1963], since this algorithm has a fast rate of convergence. A synthetic image, $\mathsf{X}[n_1, n_2]$ of $9 \times 9$ dimension, corrupted by impulsive noise of known variance $\sigma_N^2$ is simulated. Its noiseless version $\mathsf{Y_d}[n_1, n_2]$ of identical dimension is also simulated. The output

of the quadratic filter is assumed as

$$\mathsf{Y}[n_1, n_2] = \mathsf{X}^T[n_1, n_2]\mathsf{H}_{2_{\text{noise}}}\mathsf{X}[n_1, n_2] \tag{7.1}$$

Here, the expected value of the squared error between the noisy image and the noiseless version is taken as the cost function for minimization. Let the mean square error between $y_d[n_1, n_2]$ and $y[n_1, n_2]$ be $\xi$ given by

$$\xi = E[|\mathsf{Y_d}[n_1, n_2] - \mathsf{X}^T[n_1, n_2]\mathsf{H}_{2_{\text{noise}}}\mathsf{X}[n_1, n_2]|^2] \tag{7.2}$$

Minimization of $\xi$ results in an optimum $\mathsf{H}_{2_{\text{noise}}}$ and is plotted as in Fig. 7.2. The principal maxima are along the main diagonal and local maxima appear parallel to this diagonal.

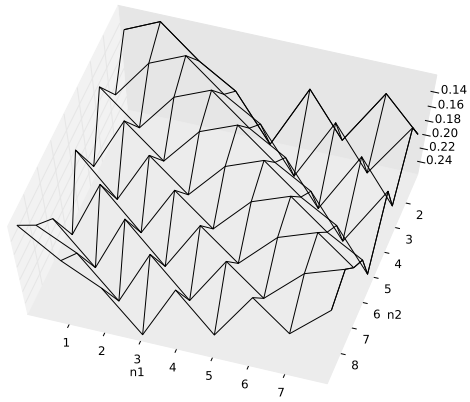



**Fig. 7.2.** Wire-frame plot of the quadratic kernel

The kernel is indicative of edge enhancement, a feature which makes this denoising filter ideal for processing raw MRI signals. The preservation of edges leads to less blur on filtering and consequently results in better demarcation of tumour tissues. A direct implementation of $\mathsf{H}_{2_{\text{noise}}}$ as in Eq. 7.1 is computationally complex. Instead, singular

value decomposition[Gantmacher 1960] is performed on $\mathsf{H}_{2_{\text{noise}}}$ to yield an approximation $\tilde{\mathsf{H}}_{2_{\text{noise}}}$ as

$$\tilde{\mathsf{H}}_{2_{\text{noise}}} = \sum_{i=1}^{\rho} \lambda_i \mathsf{S_i} \mathsf{S_i}^T \tag{7.3}$$

where $\lambda_i$ are the singular values and each $\mathsf{S_i}$ is a $9 \times 1$ eigen vector. Table 7.1 lists the eigenvalues and singular vectors of $\tilde{\mathsf{H}}_{2_{\text{noise}}}$. The value of $\rho$ is selected in such a manner that the Frobenius norm [Mathews and Sicuranza 2000] $||\mathsf{H}_{2_{\text{noise}}} - \tilde{\mathsf{H}_{2_{\text{noise}}}}||$ is minimum. Each $\mathsf{S_i}$ is re-sized as a $3 \times 3$ FIR image filter that is equivalent to $H(i,j)$ in Eq. 2.16, page 33. The outputs of FIR filters are squared and a weighted sum with $\lambda_i$ values yields the filter output. The structure of the filter is as in Fig. 5.4, page 77.

**Table 7.1:** Table of singular values and singular vectors of $\mathsf{H}_{2_{\text{noise}}}$

| $\lambda_i$ | $\mathsf{S_1}$ | $\mathsf{S_2}$ | $\mathsf{S_3}$ | $\mathsf{S_4}$ | $\mathsf{S_5}$ |
|---|---|---|---|---|---|
| 1.0670 | -0.3333 | -0.3333 | -0.3333 | -0.3333 | -0.3333 |
| 0.1735 | 0.0000 | -0.3030 | -0.4642 | -0.4082 | -0.1612 |
| 0.1735 | 0.4714 | -0.3611 | 0.0818 | -0.2357 | -0.4430 |
| 0.1627 | 0.4613 | -0.4667 | 0.4559 | -0.3146 | 0.1756 |
| 0.1627 | -0.0969 | -0.0667 | 0.4157 | -0.3151 | 0.4375 |

## 7.4 Quadratic Unsharp Masking

Unsharp masking is a contrast enhancement scheme in which the edges are filtered out, scaled and added with the image[Mitra and Sicuranza 2001]. The high pass filter enhances the edges and the addition of

edges improves the overall contrast of the image. The chief difficulty with this scheme is that the edge detection high pass filters commonly employed are very sensitive to noise. These filters respond to the false edges rendered by noise. Quadratic edge detectors are very noise immune and have better edge detection characteristics than Laplacian, Canny and LoG filters. The unsharp masking scheme employing quadratic edge detection filter is as represented in Fig. 7.3.

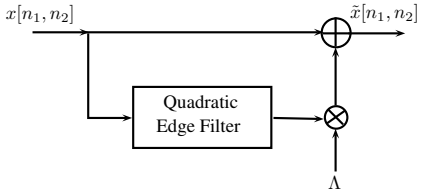



**Fig. 7.3.** Unsharp masking with quadratic filter

The edges in the input image $\mathsf{x}[n_1, n_2]$ are separated by the quadratic filter $\mathsf{H}_{2_{\text{edge}}}$. They are then scaled by a factor $\Lambda$ and are added with the input image to yield the enhanced version $\tilde{\mathsf{x}}[n_1, n_2]$. The scale factor $\Lambda$ is chosen in such a way that there is improvement in $\tilde{\mathsf{x}}[n_1, n_2]$ in respect of subjective visual quality as well as in numerical values of performance criteria like SNR, PSNR etc.

## 7.5 Design of Quadratic Edge Detection Filter

The core of the unsharp masking scheme is the quadratic edge detector that performs better than conventional edge detectors in terms of the crispness of edge and noise performance. The design follows the same strategy presented in Sec. 7.3. Here also, Powell optimization is used for obtaining $\mathsf{H}_{2_{\text{edge}}}$ [Fletcher and Powell 1963]. A blurred synthetic edge denoted as $\mathsf{x}[n_1, n_2]$ of $9 \times 9$ dimension is simulated. A

desired sharp synthetic edge $\mathsf{y_d}[n_1, n_2]$ of identical dimension is also simulated. The output of the quadratic filter, that is assumed to be $\mathsf{y}[n_1, n_2]$, follows Eq. 7.1 with $\mathsf{H}_{2_{\text{noise}}}$ replaced with $\mathsf{H}_{2_{\text{edge}}}$. The mean square error ($\xi$) between the filter output and $\mathsf{y_d}[n_1, n_2]$ is minimized to yield an optimum $\mathsf{H}_{2_{\text{edge}}}$ as plotted in Fig. 7.4.

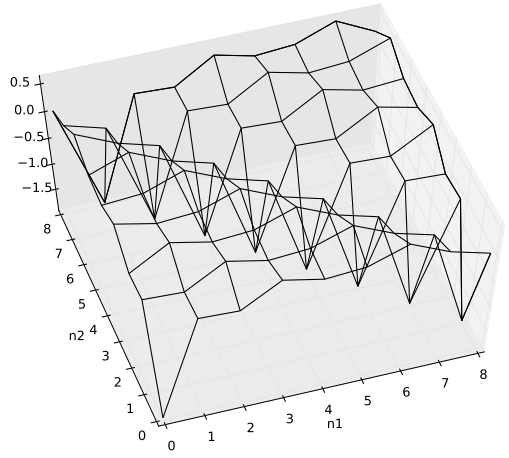



**Fig. 7.4.** Surface plot of the quadratic kernel

The kernel $\mathsf{H}_{2_{\text{edge}}}$ is not completely isotropic. It becomes zero at the coordinates $(i, N-i)$ for $1 \leq i \leq N-1$, creating a wedge like minimum parallel to the main diagonal. There are local minima parallel to this. This $\mathsf{H}_{2_{\text{edge}}}$ surface has high pass filter characteristics, making it suitable for edge detection. Once $\mathsf{H}_{2_{\text{edge}}}$ is computed, it is required to implement it as a computationally efficient structure. The direct implementation as in Eq. 2.13, page 32 is complex. Instead, singular value decomposition[Gantmacher 1960] is performed on $\mathsf{H}_{2_{\text{edge}}}$ to yield an approximation $\tilde{\mathsf{H}}_{2_{\text{edge}}}$ as in Eq. 4.24, page 63. The singular values and singular vectors in the case of $\mathsf{H}_{2_{\text{edge}}}$ are presented in Table 7.2.

**Table 7.2:** Table of singular values and singular vectors of $H_{2_{edge}}$

| $\lambda_i$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ |
|---|---|---|---|---|
| 3.2783 | -0.3333 | -0.3333 | -0.3333 | -0.3333 |
| 2.9284 | -0.0570 | 0.4337 | -0.3768 | -0.0570 |
| 2.9284 | 0.4680 | -0.1846 | -0.2833 | 0.4680 |
| 2.8494 | -0.0388 | 0.2723 | 0.4559 | 0.4262 |
| 2.8494 | -0.4698 | -0.3848 | -0.1199 | 0.2013 |
| 2.8467 | 0.0998 | -0.2513 | 0.3726 | -0.4489 |
| 2.8467 | -0.4607 | 0.3988 | -0.2888 | 0.1440 |
| 2.7422 | 0.4690 | 0.0344 | -0.4570 | -0.1931 |
| 2.7422 | -0.0478 | -0.4702 | -0.1155 | 0.4300 |
| $S_5$ | $S_6$ | $S_7$ | $S_8$ | $S_9$ |
| -0.3333 | -0.3333 | -0.3333 | -0.3333 | -0.3333 |
| 0.4337 | -0.3768 | -0.0570 | 0.4337 | -0.3768 |
| -0.1846 | -0.2833 | 0.4680 | -0.1846 | -0.2833 |
| 0.1971 | -0.1243 | -0.3875 | -0.4694 | -0.3317 |
| 0.4282 | 0.4547 | 0.2685 | -0.0434 | -0.3350 |
| 0.4711 | -0.4364 | 0.3491 | -0.2197 | 0.0638 |
| 0.0182 | -0.1783 | 0.3168 | -0.4171 | 0.4671 |
| 0.3900 | 0.3286 | -0.2759 | -0.4244 | 0.1285 |
| 0.2649 | -0.3381 | -0.3823 | 0.2053 | 0.4536 |

The ratio of the ninth eigenvalue to the first eigenvalue is significantly high. So all the nine channels are implemented, as the structure shown in Fig. 5.4, page 77. Though all the channels are employed, it does not add significantly to the complexity as the computations in all channels happen simultaneously.

The sharpness of edges is critical in MRI images because it plays a key role in the enhancement of contrast between normal tissue and the cancerous tissue. So, the unsharp masking after the removal of noise is recommended for the enhancement of peripheries in the MRI image. The high pass filter in the unsharp masking scheme is realized as a quadratic filter due to its desirable features like robustness to noise and the greater edge crispness. The experimental set up for the validation of performance parameters is detailed in Sec. 7.6.

## 7.6 Experiment

The filter kernel $\tilde{\mathsf{H}}_{2_{\mathsf{noise}}}$, designed in Sec. 7.3, is simulated in Python with the help of scipy and pylab modules. The raw MRI data is taken and the frames are separated. These noisy images are imported into Python using the image processing toolbox and subjected to filtering by $\tilde{\mathsf{H}}_{2_{\mathsf{noise}}}$. The filtered images are compared with those processed by spatial filters like median, Gaussian and minimum filters in terms of the visual quality. The experimental set up in Fig. 7.5 is used to test the noise invulnerability of the denoising filter kernel $\tilde{\mathsf{H}}_{2_{\mathsf{noise}}}$.

Impulsive noise of known variance is added with the frames and subjected to filtering. Quantitative measures like SNR and PSNR are calculated with reference to the noisy image. Experiment is repeated for conventional filters like mean, median etc. and the results are compared with those of quadratic filter. Conventional filters blur the

image while removing noise. The sharpness of edges is a key factor in demarcating a pathological disorder from normal tissue. A measure of the crispness of edges($\kappa$), as proposed by Eq. 7.4, is computed based on the noisy reference image. The results of these experiments are in Sec. 7.7. The frames filtered by the quadratic filter are then subjected to unsharp masking for improving the contrast and crispness of edges. The quadratic kernel $\tilde{\mathsf{H}}_{2_{\mathsf{edge}}}$, designed in Sec. 7.5, is implemented as in Fig. 5.4, page 77. All the nine channels are implemented as even the last eigenvalue is significant enough. This filter is then incorporated in an unsharp masking scheme as shown in Fig. 7.3. The results are detailed in Sec. 7.7.

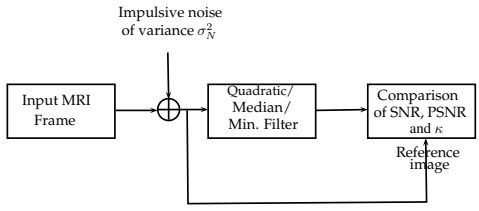
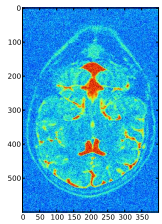



**Fig. 7.5.** Experimental set up for ascertaining the performance parameters for the noise removal filter

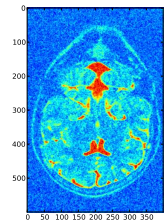
## 7.7 Results

Quadratic filter is designed and implemented with the objective of removing impulsive noise from raw MRI data which is acquired under strong and rapidly changing magnetic fields. The MRI images corrupted by impulsive noise of different noise variances are applied to the quadratic filter implemented by SVD method. The resulting output is compared with those of spatial filters like median filter, minimum filter and Gaussian filter. Fig. 7.6 shows the outputs of various

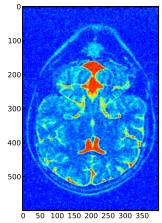
filters for impulsive noise variance $\sigma_N^2 = 100$. It is observed that quadratic filter removes the impulsive noise better than the Gaussian and median filter. The noisy MRI image is as in Fig. 7.6(a). As claimed, the output of the quadratic filter is the least noisy. Median filter and minimum filter fail to clean much of the impulsive noise in the input image. The improvement in SNR is shown in Table 7.3. There is 10 dB improvement in using quadratic filter. It has a stable noise performance even at $\sigma_N^2 = 200$ as shown in Fig. 7.7(d). Besides the improvement in SNR, quadratic filter has the advantage that the edges are not blurred on filtering, ensuring that the periphery of a possible pathological disorder like a tumour remains unambiguous.



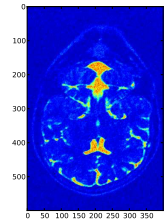
(a) Noisy MRI input



(b) Output with median filter

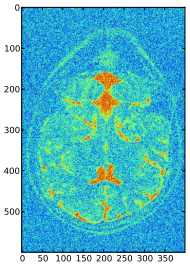


(c) Output of Gaussian filter

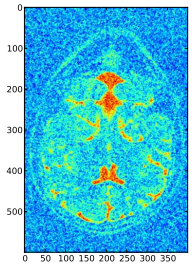


(d) Output of quadratic filter

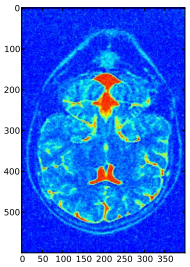
**Fig. 7.6.** Outputs of various filters for impulsive noise variance 100



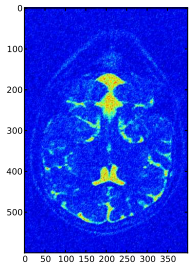

(a) Noisy MRI input




(b) Output with median filter




(c) Output of Gaussian filter




(d) Output of quadratic filter

**Fig. 7.7.** Outputs of various filters for impulsive noise variance 200

**Table 7.3:** Performance parameters for various filters

| Filter | SNR(dB) | PSNR(dB) | $\kappa$ |
|---|---|---|---|
| Quadratic | 29.76 | 35.51 | $2.48 \times 10^{-5}$ |
| Minimum | 19.30 | 22.75 | 0.0973 |
| Median | 17.48 | 20.95 | 0.0112 |

The crispness of the edges in the filtered image becomes an important criterion. A function is defined in Sec. 7.7.2 that models the crispness of edges mathematically[Thurnhofer and Mitra 1996].

### 7.7.1 Improvement in SNR and PSNR

The improvements in SNR and PSNR are computed as per the experimental setup in Fig. 7.5. The SNR is expressed as

$$SNR = 10\log_{10}\left[\frac{\sum_{n_1}\sum_{n_2} r^2[n_1,n_2]}{\sum_{n_1}\sum_{n_2}[r^2[n_1,n_2]-t^2[n_1,n_2]]}\right] \tag{7.4}$$

The peak value of the SNR is expressed as

$$PSNR = 10\log_{10}\left[\frac{\max\left(r^2[n_1,n_2]\right)}{\frac{1}{N_1N_2}\sum_{n_1}\sum_{n_2}[r^2[n_1,n_2]-t^2[n_1,n_2]]}\right] \tag{7.5}$$

where $r$ denotes the reference image and $t$ denotes the test image. $N_1N_2$ is the size of the image. Both the SNR and PSNR are direct indicators of the performance of the noise removal filter.

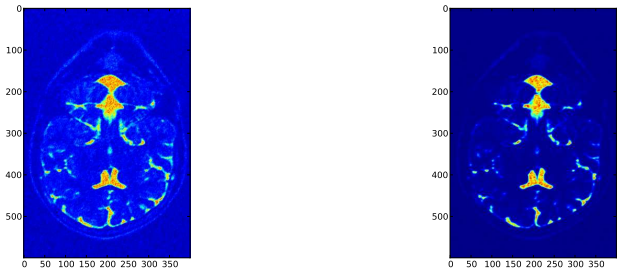
### 7.7.2 Crispness of Edges

The sharpness of edges in the filtered images is decided by a numerical figure of merit of the filter[Thurnhofer and Mitra 1996] given as

$$\kappa = \frac{1}{N_1N_2}\sum_{n_1}\sum_{n_2}\frac{|\sigma^2_{l[n_1,n_2]_{test}} - \sigma^2_{l[n_1,n_2]_{ref}}|}{\sigma^2_{l[n_1,n_2]_{ref}}\mu_{l[n_1,n_2]_{ref}}} \tag{7.6}$$

$\sigma^2_{l[n_1,n_2]_{test}}$ is the localized variance (a $3\times 3$ pixel window is used to match the size of the filter mask) of the test image and $\sigma^2_{l[n_1,n_2]_{ref}}$ is

that of the reference image. The localized mean of the reference image is $\mu_{l[n_1,n_2]_{ref}}$. One test image is formed by adding impulsive noise of variance 200 to the known MRI image to estimate the sharpness of ridges. This input image is then normalized and applied to various filters. The normalized, noisy input image is taken as the reference image and the normalized outputs of various filters are taken as test images. The variance and mean values are computed over a $3 \times 3$ pixel mask and summation is done all over the area of image. The parameter $\kappa$ should be zero when no noise is present and it increases monotonically as degradation of edges by noise increases. Table 7.3 summarizes the values of $\kappa$. The performance parameter is the lowest with quadratic filter, indicating its edge preserving feature. The small reduction in contrast on filtering is compensated by unsharp masking as depicted in Fig. 7.3. The output of the quadratic unsharp masking scheme that receives an input MRI image corrupted by impulsive noise of variance 100 is as shown in Fig. 7.8.




(a) Noisy MRI input (b) Unsharp masked output

**Fig. 7.8.** Contrast enhancement of filtered frames by unsharp masking

### Computational Complexity

The principal critique against Volterra quadratic systems is the enormous complexity arising from Kronecker products when a direct implementation is resorted to. This difficulty is overcome by the approximation of filter kernel by singular value decomposition. It is necessary to ascertain the complexity of computation on employing the quadratic filter based on singular value decomposition and then to compare it with conventional nonlinear filters. Table 7.4 shows the times of computation for various filters using Python on a core i3 Intel processor with identical image inputs to all filters. The table shows that the complexity of quadratic filters is slightly higher than that by median, Gaussian and mean filters. But the 10 dB improvement in signal to noise ratio overrides this minor demerit.

**Table 7.4:** Time of computation in seconds for various image filters

| Image size (pixels) | Median | Gaussian | Minimum | Quadratic (SVD) |
|---|---|---|---|---|
| $650 \times 1105$ | 0.19 | 0.06 | 0.03 | 1.65 |

## Conclusion

The chapter summarizes the design and implementation of a quadratic noise removal filter based on Volterra series. The quadratic kernel designed by optimization method is subjected to singular value decomposition to yield an approximate but computationally simple implementation. The filter offers 10 dB improvement in SNR with better edge preserving features than conventional filters that renders it ideal for processing MRI images. The slight reduction in contrast on filtering

is compensated by unsharp masking with a quadratic edge detection filter. The quadratic filtering operation followed by unsharp masking is observed to remove the impulsive noise present in raw MRI data much better than the conventional spatial filtering methods.

# Part IV

# Quadratic Filters for Prediction of Signals

# Foreword to Part IV

Speech coding is an important application of signal processing. Conventional methods are broadly classified into

- Temporal waveform coding
- Spectral waveform coding and
- Model based coding

The first method has gained popularity over others. Pulse code modulation(PCM), differential pulse code modulation(DPCM), delta modulation etc. fall into this category. The focus of the present work is on differential pulse code modulation, with the linear predictor in it augmented or replaced using a quadratic predictor, with the objective of modeling the nonlinear components arising from the speech generation mechanisms. The existing linear predictor based on Weiner-Hopf equations and lattice type quadratic predictor based on minimum mean square error method are implemented as benchmark systems. Optimization based predictor is designed and realized. The performance parameters are compared with the former two methods. All the three predictors are incorporated into differential pulse code modulation systems that transmit and receive speech samples. The fidelity of reception are compared among the three methods.

Chapter 8

# Differential Speech Coders based on Quadratic Predictors

## Contents

## 8.1 Introduction

The linear pulse code modulation (PCM), widely used in digital telephony [Bellamy 2000], is a memoryless coding method[Proakis 2001] that converts each speech sample into a binary code. It relies on a uniform sampler, linear quantizer and encoder for this conversion. It has the disadvantage that the statistical correlation between successive speech samples is not utilized, resulting in long binary codes. The length of the code can be effectively reduced if a differential pulse coding system (DPCM) is used, which effectively encodes the difference between each sample and its predicted value, computed based on the statistics of $N$ previous samples. There is a uniform sampler, logarithmic quantizer and a linear predictor at the core of this differential system. The reliability of the predictor is critical as prediction errors cumulatively affect successive samples on transmission and reception.

Prediction is an important signal processing operation that involves predicting the next value of a random variable or random process from the past $N$ samples. Often, linear FIR filters that are modeled based on the knowledge of the second order statistics of the input signal is employed for this purpose [Oppenheim and Schafer 1998],[Proakis and Manolakis 1998]. Although they are simple in design and structure, they fail to account for the inherent nonlinearities in the signal. This warrants the need for nonlinear processing.

In this work, Volterra series is invoked for processing the nonlinear components. It is a power series with a constant as the first term. The second term models the linear relationship between input and output, equivalent to the LTI system. The third term in the series

models the quadratic nonlinearity; the fourth term models the cubic nonlinearity and so on. So this series has the added advantage that existing LTI systems can be augmented by adding parallel polynomial filters to achieve improved performance. It is widely observed that much of the nonlinear behavior can be modeled with the quadratic term alone.

The idea is extended to the development of a quadratic predictor that works in conjunction with a linear predictor for predicting speech samples in a differential pulse code modulation (DPCM) system. The conventional linear predictor employed for this purpose cannot account for the polynomial product terms in the speech signal arising from multiple reflections in the vocal tract. Nonlinear components also arise from the harmonic distortion [Abuelmatti 1990] in microphones. The quadratic filter that acts in parallel with the linear predictor gives improved mean square error (MSE) between the actual signal and the predicted value. The differential PCM system that incorporates quadratic predictors is discussed in the subsequent sections.

## 8.2 Methodology

The analysis of performance of differential speech coding methods based on quadratic predictors is done as shown in Fig. 8.1. The objective is to design and implement a DPCM system incorporating a quadratic predictor and to compare its performance with DPCM systems employing linear predictor and lattice type quadratic predictor. In the first phase of the work, three types of predictors *viz.*

- Lattice type linear predictor based on Weiner-Hopf equations
- Lattice type quadratic predictor based on MMSE method

- Quadratic predictor based on optimization method

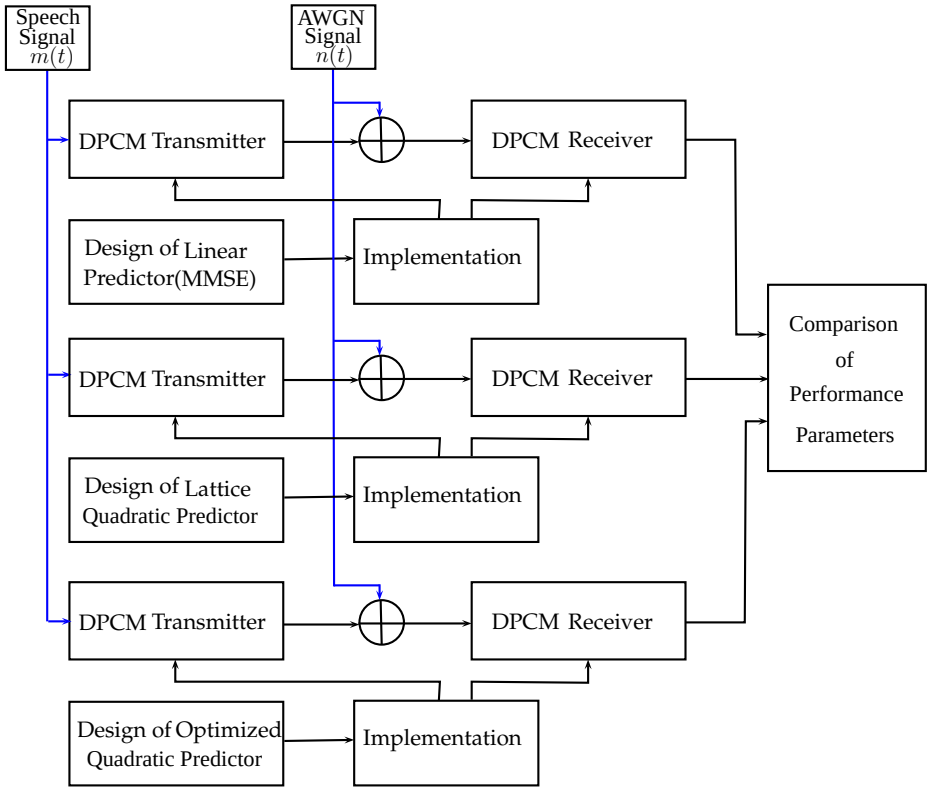



**Fig. 8.1.** Scheme of work in differential speech coding

are designed, implemented and tested with speech signal input samples. Some of the existing works are to be repeated here for effecting the comparison of the proposed system with them. The details of the predictors that are reworked are outlined below.

- *The first predictor is currently used for speech prediction, the performance of which is taken as the referenceProakis [2001].*

- *The design of the lattice type quadratic predictor, extensively used for Gaussian signals, is included as Sec. 8.4. Being less popular, its design steps are detailed in Sec. 8.4 with minor modifications, to suit the use with speech signals. More number of autocorrelation coefficients are considered in the design with speech signals than that with Gaussian signals.*

In the second phase of the work, the linear predictor designed is included in DPCM transmitter. Speech signal from a microphone is transmitted over an additive white Gaussian noise (AWGN) channel. The signal out of the channel is received by a DPCM receiver which employs an identical linear predictor. The performance parameters like SNR and mean square error are computed for various levels of channel noise variance.

In the third phase of the work, lattice type quadratic predictor based on minimum mean square error (MMSE) is included in parallel with the linear predictor of the DPCM system implemented. The performance parameters are recomputed for the same levels of channel noise. In the final phase of work, the kernel of the quadratic predictor designed by optimization method is included in the DPCM system. The performance parameters are then recomputed and compared with other methods.

## 8.3 Design and Implementation of Linear Predictor

The first known statistical prediction was made on sunspot cycles [Yule 1927] by Yule whose idea was later extended by Walker in his paper [Walker 1931]. The mathematical model presented by Yule and Walker are employed in the design of linear predictor for speech signals. Such

a predictor is essentially an FIR filter described by the equation

$$y_n = \sum_{i=0}^{N-1} a_i x_{n-i} \tag{8.1}$$

The coefficients $a_i$ are obtained by solving the Yule-Walker equations

$$\sum_{k=0}^{p} a(l)\phi_{xx}(l-k) = 0; \quad l = 1, 2, \ldots, p \tag{8.2}$$

by Levinson-Durbin recursion[Proakis 2001]. Here $\phi_{xx}$ is the auto-correlation matrix of $x$. Such linear systems are not suitable for processing nonlinearities in the input $x$. Most real life systems and processes are nonlinear, do not obey superposition. The common examples from engineering fields being saturation, hysteresis etc. The biggest challenge in non linear signal processing is the computational complexity. Nevertheless, the existing methods and algorithms for LTI systems can be extended to nonlinear systems. Recently, with added computing power, focus has been shifted to modeling and processing nonlinearities. The task is to design and realize a Volterra series based nonlinear predictor for speech signals which inherently contain polynomial components due to multiple reflections in the vocal tract. The linear predictor with a lattice structure is as shown in Fig. 8.2. It has a compact modular structure, the design of which is in finding the reflection coefficients $K_i$. An iterative solution for $K_i$ is possible using Levinson-Durbin recursion.

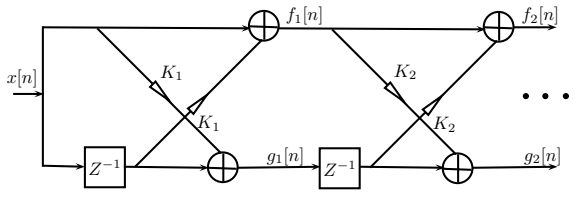



**Fig. 8.2.** Block diagram of lattice linear predictor

The linear predictor derived by LD recursion gives the best performance for Gaussian input signal as shown in Fig. 8.3. Its performance degrades for speech signal as shown in Fig. 8.4. The mean square error between the actual signal and the predicted signal can be reduced if one employs a quadratic lattice predictor. The design steps as adapted from [Mathews and Sicuranza 2000] are discussed in the next section.

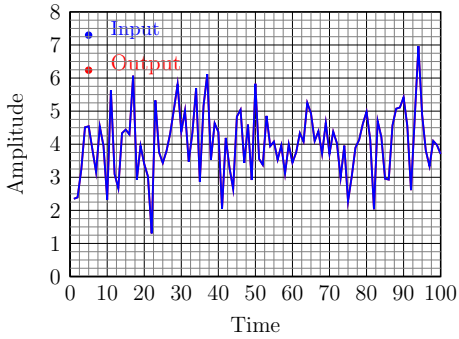



**Fig. 8.3.** Output of linear predictor for random Gaussian input

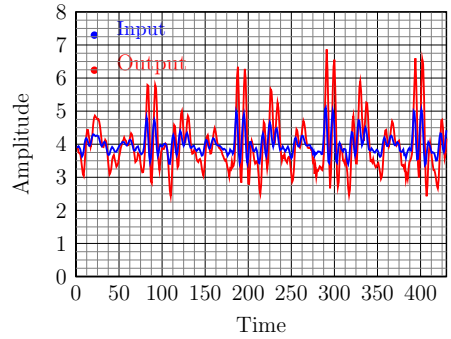



**Fig. 8.4.** Output of linear predictor for speech signal input

## 8.4 Design and Implementation of Lattice type Quadratic Predictor

Quadratic product components can be added to the linear lattice filter to model polynomial components in the speech signal. Block diagram of such a lattice type quadratic predictor is shown in Fig. 8.5. It differs from the linear lattice filter in three aspects.

- The reflection coefficients are no longer scalars, but matrices whose dimensions increment with every stage of computation.

- It is not modular.
- The data sets, that each stage operates on, follow a pattern as in Fig. 8.6.

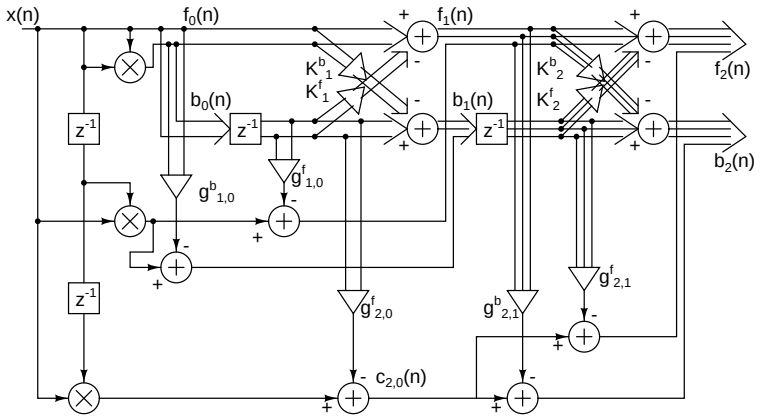



**Fig. 8.5.** Quadratic lattice predictor

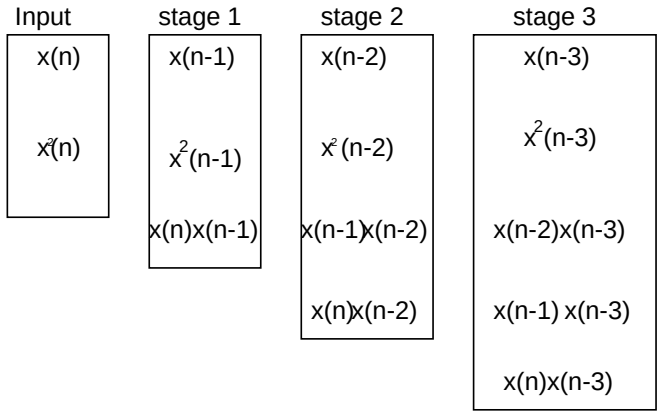



**Fig. 8.6.** Signal sets used in every stage

A three stage lattice with quadratic nonlinearity is considered. The basic idea employed in the derivation is very similar to the lattice orthogonalization[Mathews and Sicuranza 2000] using linear prediction techniques. The input to the first stage is the data vector with linear and quadratic components given as $\mathsf{x_0}^{\mathsf{b}}[n] = [\, x[n] \quad x^2[n] \,]^T$. The second stage operates on the delayed version appended with the cross product term $x[n]x[n-1]$ expressed as a vector with three components as $\mathsf{x}_1^{\mathsf{b}}[n] = [\, x[n-1] \quad x^2[n-1] \quad x[n]x[n-1] \,]^T$.
Three vectors are defined as

$$\mathsf{x}_0^{\mathsf{b}}[n] = [\, x[n] \quad x^2[n] \,]^T \tag{8.3}$$

$$\mathsf{x}_1^{\mathsf{b}}[n] = [\, x[n-1] \quad x^2[n-1] \quad x[n]x[n-1] \,]^T \tag{8.4}$$

and

$$\mathsf{x}_2^{\mathsf{b}}[n] = [\, x[n-2] \quad x^2[n-2] \quad x[n-1]x[n-2] \quad x[n]x[n-2] \,]^T \tag{8.5}$$

Note that

$$\mathsf{x}_1^{\mathsf{b}}[n] = [\, (\mathsf{x}_0^{\mathsf{b}}[n-1])^T \quad x[n]x[n-1] \,]^T \tag{8.6}$$

and

$$\mathsf{x}_2^{\mathsf{b}}[n] = [\, (\mathsf{x}_1^{\mathsf{b}}[n-1])^T \quad x[n]x[n-2] \,]^T \tag{8.7}$$

The vector $\mathsf{x}_1^{\mathsf{b}}[n]$ and $\mathsf{x}_2^{\mathsf{b}}[n]$ are recognized as

$$\hat{\mathsf{x}}_1^{\mathsf{b}}[n] = \mathsf{B}_{01}\mathsf{x}_0^{\mathsf{b}}[n] \tag{8.8}$$

and

$$\mathsf{B}_{01} = (E\{\mathsf{x}_0^{\mathsf{b}}[n]\mathsf{x}_0^{\mathsf{b}}[n]^T\})^{-1} E\{\mathsf{x}_1^{\mathsf{b}}[n](\mathsf{x}_0^{\mathsf{b}}[n])^T\} \tag{8.9}$$

Eq. 8.9 is similar to Wiener-Hopf equation as in the case of linear predictor, except that $\mathsf{B}_{01}$ is matrix rather than a scalar value. The forward reflection coefficient matrix for the first stage is given as

$$\mathsf{K}_1^{\mathsf{b}} = \mathsf{B}_{01} \tag{8.10}$$

and the backward reflection coefficient matrix for the first stage is given as

$$\mathsf{K}_1^{\mathsf{f}} = \mathsf{B}_{01} \tag{8.11}$$

It is needed to estimate $x[n]x[n-1]$ using $[\,x[n], x^2[n]\,]^T$. The corresponding coefficient vector is

$$\mathsf{g}_{1,0}^{\mathsf{b}} = \left( E\left\{ \begin{bmatrix} x[n] \\ x^2[n] \end{bmatrix} [\,x[n] \;\; x^2[n]\,] \right\} \right)^{-1} E\left\{ x[n]x[n-1] \begin{bmatrix} x[n] \\ x^2[n] \end{bmatrix} \right\} \tag{8.12}$$

$$\mathsf{g}_{1,0}^{\mathsf{f}} = \left( E\left\{ \begin{bmatrix} x[n-1] \\ x^2[n-1] \end{bmatrix} [\,x[n-1] \;\; x^2[n-1]]\right\} \right)^{-1} \times E\left\{ x[n]x[n-1] \begin{bmatrix} x[n-1] \\ x^2[n-1] \end{bmatrix} \right\} \tag{8.13}$$

$$E\{\mathsf{b}_1[n]{\mathsf{b}_1}^T[n]\} = E\{\mathsf{X}_1^{\mathsf{b}}[n]\mathsf{X}_1^{\mathsf{b}}[n])^T\} - \begin{bmatrix} (\mathsf{K}_1^{\mathsf{b}})^T \\ (\mathsf{g}_{1,0}^{\mathsf{b}})^T \end{bmatrix} E\{\mathsf{X}_0^{\mathsf{b}}[n](\mathsf{x}_1^{\mathsf{b}}[n])^T\} \tag{8.14}$$

$$E\left\{ \mathsf{b}_1[n-1] \begin{bmatrix} x[n] \\ x^2[n] \\ x[n]x[n-1] \end{bmatrix}^T \right\} =$$

$$E\left\{ \left( \begin{bmatrix} x[n-2] \\ x^2[n-2] \\ x[n-1]x[n-2] \end{bmatrix} - \begin{bmatrix} (\mathsf{K}_1^{\mathsf{b}})^T \\ (\mathsf{g}_{1,0}^{\mathsf{b}})^T \end{bmatrix} \begin{bmatrix} x[n-1] \\ x^2[n-1] \end{bmatrix} \right) \begin{bmatrix} x[n] \\ x^2[n] \\ x[n]x[n-1] \end{bmatrix}^T \right\} \quad (8.15)$$

The matrix of forward reflection coefficient for the second stage is expressed as

$$\mathsf{K}_2^{\mathsf{f}} = (E\{\mathsf{b}_1[n]{\mathsf{b}_1}^T[n]\})^{-1} E\left\{ \mathsf{b}_1[n-1] \begin{bmatrix} x[n] \\ x^2[n] \\ x[n]x[n-1] \end{bmatrix}^T \right\} \quad (8.16)$$

$$E\{\mathsf{f}_1[n]{\mathsf{f}_1}^T[n]\} = E\{\mathsf{b}_1[n]{\mathsf{b}_1}^T[n]\} \quad (8.17)$$

The matrix of backward reflection coefficient for the second stage is expressed as

$$\mathsf{K}_2^{\mathsf{b}} = (E\{\mathsf{f}_1[n](\mathsf{f}_1{}^T[n]\})^{-1} E\left\{ \mathsf{f}_1[n] \begin{bmatrix} x[n-2] \\ x^2[n-2] \\ x[n-1]x[n-2] \end{bmatrix}^T \right\} \quad (8.18)$$

$$E\{\mathsf{b}_1[n-1]\ x[n]x[n-2]\} = E\left\{\left(\begin{bmatrix} x[n-2] \\ x^2[n-2] \\ x[n-1]x[n-2] \end{bmatrix} - \begin{bmatrix} (\mathsf{K}_1^{\mathsf{b}})^T \\ (\mathsf{g}_{1,0}^{\mathsf{b}})^T \end{bmatrix} \begin{bmatrix} x[n-1] \\ x^2[n-1] \end{bmatrix}\right) x[n]x[n-2]\right\} \tag{8.19}$$

$$\mathsf{g}_{2,1}^{\mathsf{f}} = (E\{\mathsf{b}_1[n-1](\mathsf{b}_1{}^T[n-1]\})^{-1} E\{\mathsf{b}_1[n-1]\ x[n]x[n-2]\} \tag{8.20}$$

$$E\{\mathsf{f}_1[n]\ x[n]x[n-2]\} = E\left\{\left(\begin{bmatrix} x[n] \\ x^2[n] \\ x[n]x[n-1] \end{bmatrix} - \begin{bmatrix} (\mathsf{K}_1^{\mathsf{f}})^T \\ (\mathsf{g}_{1,0}^{\mathsf{f}})^T \end{bmatrix} \begin{bmatrix} x[n-1] \\ x^2[n-1] \end{bmatrix}\right) x[n]x[n-2]\right\} \tag{8.21}$$

$$\mathsf{g}_{2,1}^{\mathsf{b}} = (E\{\mathsf{f}_1[n](\mathsf{f}_1{}^T[n]\})^{-1} E\{\mathsf{f}_1[n]\ x[n]x[n-2]\} \tag{8.22}$$

$\mathsf{f}_0[n]$ and $\mathsf{b}_0[n]$ are defined as:

$$\mathsf{f}_0[n] = \mathsf{b}_0[n] = \begin{bmatrix} x[n] \\ x^2[n] \end{bmatrix} \tag{8.23}$$

$$\mathsf{b}_1[n] = \begin{bmatrix} \mathsf{b}_0[n-1] - (\mathsf{K}_1^{\mathsf{b}})^T \mathsf{f}_0[n] \\ x[n]x[n-1] - (\mathsf{g}_{1,0}^{\mathsf{b}})^T \mathsf{f}_0[n] \end{bmatrix} \tag{8.24}$$

$$\mathsf{f}_1[n] = \begin{bmatrix} \mathsf{f}_0[n] - (\mathsf{K}_1^{\mathsf{f}})^T \mathsf{b}_0[n-1] \\ x[n]x[n-1] - (\mathsf{g}_{1,0}^{\mathsf{f}})^T \mathsf{b}_0[n-1] \end{bmatrix} \tag{8.25}$$

$$\mathsf{c}_{2,0}[n] = x(n)x(n-2) - (\mathsf{g}_{2,0}^{\mathsf{f}})^T \mathsf{b}_0[n-1] \tag{8.26}$$

$$\mathsf{b}_2[n] = \begin{bmatrix} \mathsf{b}_1[n-1] - (\mathsf{K}_2^{\mathsf{b}})^T \mathsf{f}_1[n] \\ \mathsf{c}_{2,0}[n] - (\mathsf{g}_{2,1}^{\mathsf{b}})^T \mathsf{f}_1[n] \end{bmatrix} \tag{8.27}$$

$$\mathsf{f}_2[n] = \begin{bmatrix} \mathsf{f}_1[n] - (\mathsf{K}_2^{\mathsf{f}})^T \mathsf{b}_1[n-1] \\ \mathsf{c}_{2,0}[n] - (\mathsf{g}_{2,1}^{\mathsf{f}})^T \mathsf{b}_1[n-1] \end{bmatrix} \tag{8.28}$$

$$c_{i,0}(n) = x(n)x(n-i) - (\mathsf{g}_{\mathsf{i},0}^{\mathsf{f}})^T \mathsf{b}_0[n-1]; \quad i = 2, 3, \cdots, N-1 \tag{8.29}$$

The above steps are to be repeated for $i = 3, 4, \cdots, N-1$
Backward prediction error update:

$$\mathsf{b}_{\mathsf{i}}[n] = \begin{bmatrix} \mathsf{b}_{\mathsf{i}-1}[n-1] - (\mathsf{K}_{\mathsf{i}}^{\mathsf{b}})^T \mathsf{f}_{\mathsf{i}-1}[n] \\ c_{i,i-2}(n) - (\mathsf{g}_{\mathsf{i},\mathsf{i}-1}^{\mathsf{b}})^T \mathsf{f}_{\mathsf{i}-1}[n] \end{bmatrix} \tag{8.30}$$

Forward prediction error update:

$$\mathsf{f}_{\mathsf{i}}[n] = \begin{bmatrix} \mathsf{f}_{\mathsf{i}-1}[n] - (\mathsf{K}_{\mathsf{i}}^{\mathsf{f}})^T \mathsf{b}_{\mathsf{i}-1}[n-1] \\ c_{i,i-2}[n] - (\mathsf{g}_{\mathsf{i},\mathsf{i}-1}^{\mathsf{f}})^T \mathsf{b}_{\mathsf{i}-1}[n-1] \end{bmatrix} \tag{8.31}$$

Auxiliary variable update:

$$c_{j,i-1}[n] = c_{j,i-2}[n] - (\mathsf{g}^{\mathsf{f}}_{\mathsf{j},\mathsf{i}-1})^T \mathsf{b}_{\mathsf{i}-1}[n-1]; \quad j = i+1, i+2, \cdots, N-1 \tag{8.32}$$

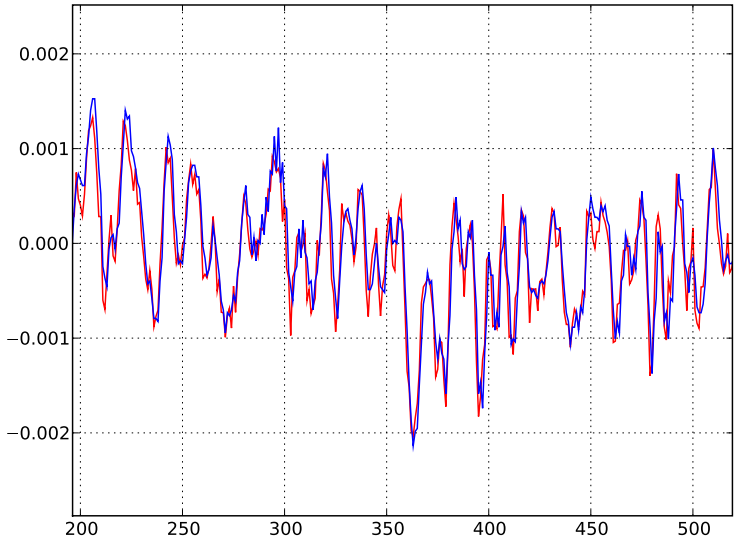



**Fig. 8.7.** Output of the lattice predictor for speech signal input

The quadratic predictor so designed is implemented and tested with speech samples. The output is as shown in Fig. 8.7 which indicates less mean square prediction error than the linear predictor designed in Sec. 8.3. But it suffers from large computational complexity. So an approximate implementation of the quadratic kernel as in Eq. 2.6 is sought for. The strategy adopted in the first method is to incorporate a quadratic predictor in parallel with a linear predictor to augment

the performance of the latter. The second method dispenses the linear predictor and relies on a quadratic predictor alone. The design methodology is detailed in the following sections.

## 8.5 Quadratic predictor based on Optimization

A quadratic filter based on optimization method is used to replace the linear predictor to yield improved performance. Let $\mathsf{Y}[n]$ be the actual output signal and $\hat{\mathsf{Y}}[n]$ be its predicted value rendered by a quadratic system as

$$\hat{\mathsf{Y}}[n] = \mathsf{X}^T[n]\mathsf{H}_2\mathsf{X}[n] \tag{8.33}$$

An objective function $J(H)$ is defined as

$$J(H) = E[\,|\mathsf{Y}[n] - \mathsf{X}^T[n]\mathsf{H}_2\mathsf{X}[n]|^2\,] \tag{8.34}$$

The function $J(H)$ is minimized using Powell method to yield the optimum $\mathsf{H}_{2_{\mathsf{opt}}}$ which is plotted as shown in Fig. 8.8.

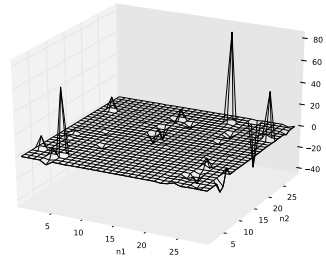



**Fig. 8.8.** Plot of $\mathsf{H}_{2_{\mathsf{opt}}}$

The matrix $\mathsf{H}_{2_{\mathsf{opt}}}$ is composed of four prominent impulses. Being one dimensional processing, the computational challenges are less and a

direct implementation as in Eq. 2.10 is employed for quadratic predictor.

## 8.6 Differential Pulse Code Modulation System with Quadratic Predictor

Differential Pulse Code Modulation (DPCM) is a variant of the linear PCM. In linear PCM, the analog speech signal is sampled above the Nyquist rate [Nyquist 2002] and each sample is quantized and encoded. Long code words result as the correlation between successive speech samples is not exploited. The statistical dependence between samples can be effectively utilized in a differential scheme in which the difference between the present sample and its predicted value is quantized and encoded. Since the difference between successive samples is much smaller than sample value itself, the quantizer need to have a logarithmic characteristics and the length of the code is smaller than in linear PCM. The core part of the DPCM system is the predictor that models the sample value.

Conventional DPCM system uses a linear predictor, designed based on Weiner-Hopf equations, for estimating the sample value[Proakis 2001]. The linear predictor does not account for the nonlinear components in the speech signal. These polynomial components can be modeled by adding a quadratic predictor in parallel with the linear predictor as in Eq. 2.6, thereby improving the signal fidelity. The existent DPCM system is modified by inserting a quadratic predictor in parallel with the linear predictor as shown in Fig. 8.9 and Fig. 8.10. The predicted value of the speech signal, supplied by the linear and quadratic predictor jointly, is subtracted from the signal itself and the resulting error signal is logarithmically quantized in the transmitter. The quantized

error samples are encoded. Both predictors work in a feedback loop to estimate the speech signal from the output and the quantized error signal. There is an identical quadratic predictor working in parallel with the linear predictor at the receiving end. The quadratic predictors at both ends are the novelty in the work. Two types of quadratic predictors, used in the modified DPCM system, are designed, realized and their performance parameters are assessed with the linear predictor as the benchmark.

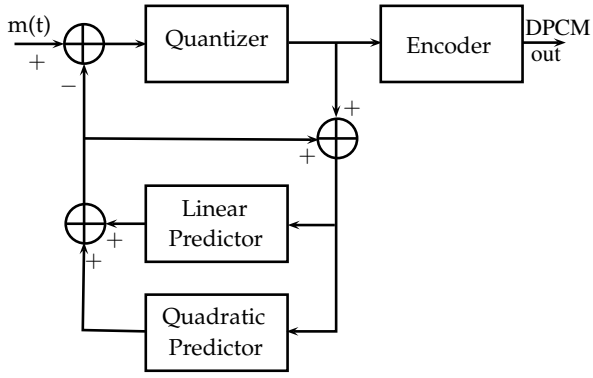



**Fig. 8.9.** DPCM transmitter with quadratic predictor

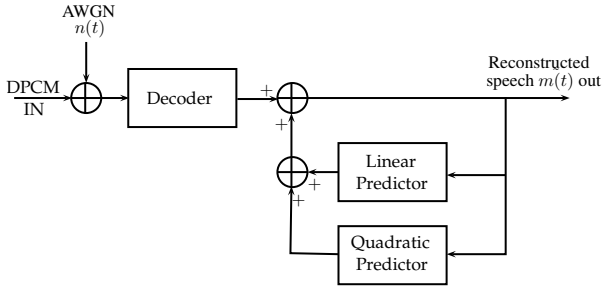



**Fig. 8.10.** DPCM receiver with quadratic predictor

## 8.7 Experiment

The experiment is designed with the objective of assessing the quality of the received speech signal from a DPCM transmitter - receiver based on an optimized quadratic predictor. Comparison is performed between differential coding systems employing the conventional linear predictor and the lattice type quadratic predictor. A quantitative analysis of the linear and quadratic predictors implemented as part of the DPCM system is done in two phases. In the first phase, the predictors are tested with random Gaussian signals and speech signals. The mean square error between actual signal and predicted signal is ascertained in each case.

In the second phase of experiment, differential pulse code modulation system shown in Fig. 8.9 and in Fig. 8.10 are realized and the performance parameters are tested with speech signals with controlled amount of white Gaussian noise.

## 8.8 Results

Quadratic predictors, both optimization based and lattice type, are tested with speech signals and then included in the modified DPCM system. The output of DPCM system incorporating a linear predictor for Gaussian noise variance 2500 and 40000 are shown in Figs. 8.11 and 8.12 respectively.

The blue waveform corresponds to the transmitted signal and the red waveform corresponds to the received signal. The absolute error between the two is prominent for low and high frequency components and it increases as the noise variance increases. The error signal for

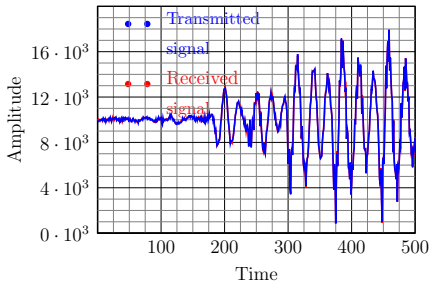



**Fig. 8.11.** Received and transmitted signals for $\sigma_N^2 = 2500$ with linear lattice predictor

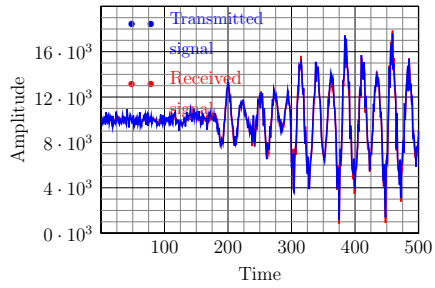



**Fig. 8.12.** Received and transmitted signals for $\sigma_N^2 = 40000$ with linear lattice predictor

$\sigma_N 2 = 2500$ is shown in Fig. 8.13. The absolute error increases as the noise variance increases to 40000 as shown in Fig. 8.14.

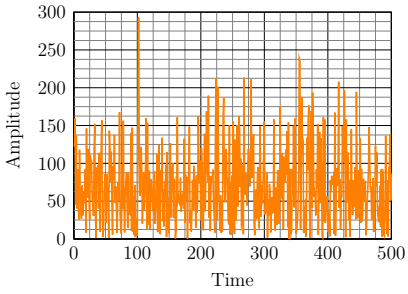



**Fig. 8.13.** Error signal with linear predictor for $\sigma_N^2 = 2500$

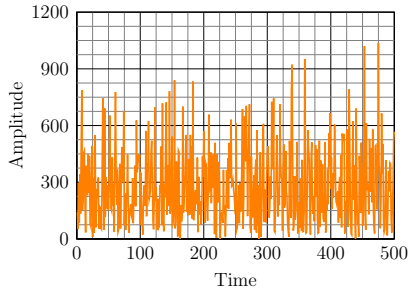



**Fig. 8.14.** Error signal with linear redictor for $\sigma_N^2 = 40000$

The absolute error between The transmitted and received signal, with quadratic lattice predictor and linear predictor working jointly, for Gaussian noise variance 2500 and 40000 are shown in Figs. 8.15 and 8.16 respectively. The error between them is less than that with a linear predictor but the overshoots are present at high frequencies.

Figs. 8.17 and 8.18 shows the error signal in each case.

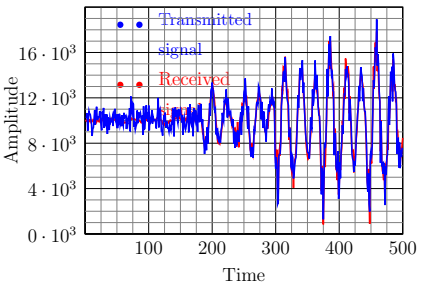



**Fig. 8.15.** Received and transmitted signals for $\sigma_N^2 = 2500$ with quadratic lattice predictor

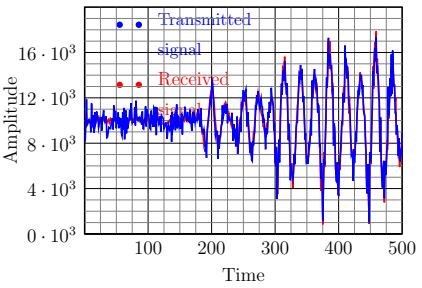



**Fig. 8.16.** Received and transmitted signals for $\sigma_N^2 = 40000$ with quadratic lattice predictor

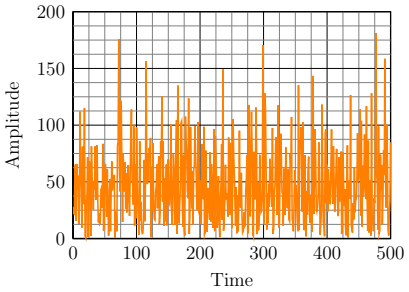



**Fig. 8.17.** Absolute error for for $\sigma_N^2 = 2500$ with quadratic lattice predictor

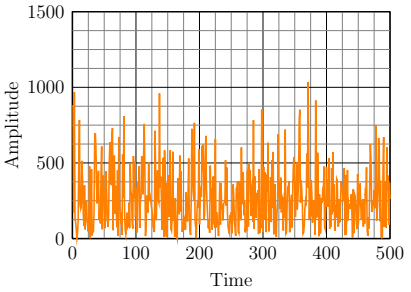



**Fig. 8.18.** Received and transmitted signals for $\sigma_N^2 = 40000$ with quadratic lattice predictor

Figs. 8.19 and 8.20 show the transmitted and received signal with optimization based quadratic predictor alone. The error between transmitted and received signal is the smallest in this case and it does not increase substantially as the noise variance increases, as shown in Figs. 8.21 and 8.22. Both the low frequency and the high frequency

components are faithfully reproduced.

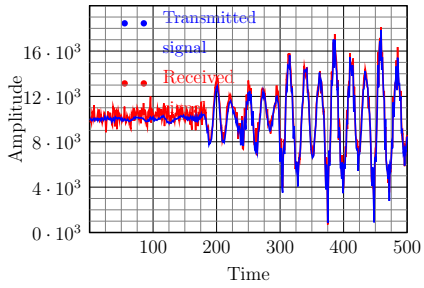



**Fig. 8.19.** Received and transmitted signals for $\sigma_N^2 = 2500$ with quadratic predictor based on optimization

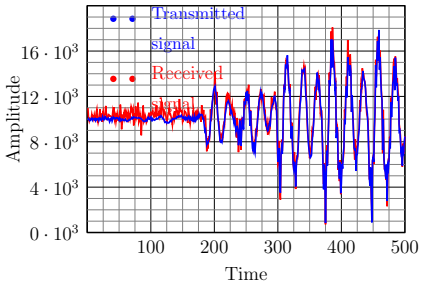



**Fig. 8.20.** Received and transmitted signals for $\sigma_N^2 = 40000$ with quadratic predictor based on optimization

The visual observations based on waveforms may not be very informative. Distinction is clearly made on hearing the audio output. Apart from these two validations, comparison of the mean square error between transmitted and received signals using the three predictors is done as another step of performance validation. If the input speech sample $x$ is composed of $N$ frames of $x_i$ and the received signal($y$) contains $N$ frames $y_i$, then the mean square error is computed as

$$MSE = \frac{1}{N} \sum_{i=1}^{N} E[|x_i - y_i|^2] \tag{8.35}$$

The mean square error is plotted in Fig. 8.23. In every case, the mean square error follows a square law relationship with noise variance, meaning that the absolute error between the transmitted and

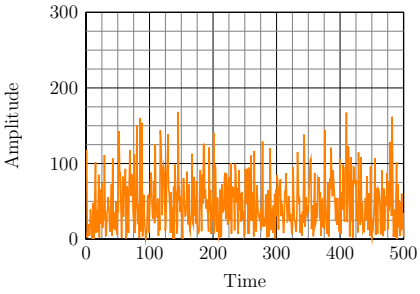



**Fig. 8.21.** Received and transmitted signals for $\sigma_N^2 = 2500$ with quadratic predictor based on optimization

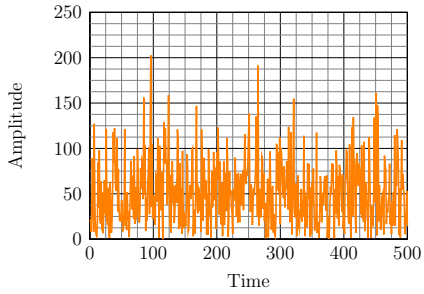



**Fig. 8.22.** Received and transmitted signals for $\sigma_N^2 = 40000$ with quadratic predictor based on optimization

received signal increases linearly with the noise variance. The average squared error with quadratic lattice filter is less than half that with linear predictor. The MSE in the case of the optimization based quadratic filter is one third that of the linear predictor.

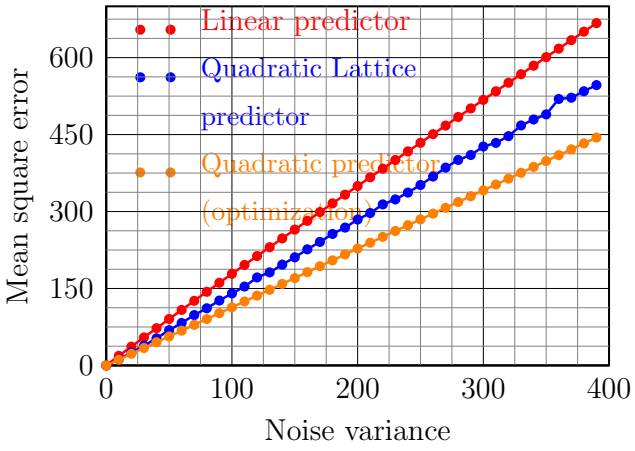



**Fig. 8.23.** Mean square error with different predictors

## 8.9 Conclusion

Differential pulse code modulation is the source coding system that exploits the correlation between input sample and the previous $N$ samples. A predictor is the core part of the transmitter and receiver which estimates the next sample based on the knowledge of the past $N$ samples. The traditional linear predictor based on Wiener-Hopf equation cannot account the polynomial components in speech signal arising from multiple reflections in the vocal tract. The work proposes the use of a quadratic predictor in place of the linear predictor. A three stage quadratic lattice predictor designed based on minimum mean square error is implemented and tested with random Gaussian signal and speech signal. It is incorporated in the DPCM system and the performance parameters are contrasted with one employing linear predictor. Though the performance in terms of MSE is better it suffered from overshoots at high frequencies. To resolve this problem, a quadratic predictor designed based on the minimization of mean square error is designed, tested and used in the DPCM system. The new system yielded the lowest mean square error among the three and interpolated the speech signal irrespective of the frequency content and the Gaussian noise variance.

# Part V

# Summary, Inferences and Research Contributions

# Foreword to Part V

The previous chapters presented the motivation and the research work done on quadratic systems for three practical applications. Chapter 9 in part V concludes the research work, highlighting the achievements in employing quadratic filters for each area of application. The improved performance parameters in the case of quadratic filters are demarcated by bar charts. Some limitations that a system designer faces when working with quadratic systems are presented in this chapter, besides the advantages. Future scope for the work and possible expansions follow this.

The research contributions and their impacts to various stakeholders are detailed in chapter 10. The list of publications are appended to this chapter, which includes both the published and communicated papers in journals and international conferences.

Chapter 9

# Summary and Inferences

## Contents

## 9.1 Overview

Quadratic filters for edge detection, denoising and prediction are presented in the last part. Their performance indicators are compared with existing systems to infer the merits of quadratic systems. Sec. 9.2 gives a bird's eye view of the whole research work highlighting the numerical figures of merit of quadratic systems. Sec. 9.3 presents the limitation in developing quadratic filters. The future scope and possible expansions are discussed in Sec. 9.4.

## 9.2 Summary of Work Done

The research work is primarily conceived with the notion that mild polynomial nonlinearities can be modeled using Volterra series. The filters based on this power series can outperform conventional linear and nonlinear systems, especially when there are polynomial components present in the signals processed. Three avenues, where effects of nonlinearity are strong and consequently the usefulness of quadratic systems are high, are selected as prediction, edge detection and noise removal. The crux of the work is the design and implementation of quadratic filters for these applications as outlined in the next sections.

### 9.2.1 Strategy of Design and Implementation

Quadratic filters cannot easily be designed by conventional methods. They are designed by utilizing the responses due to strategically placed bi-impulses or by optimization methods. Optimization of a cost function is selected as the design tool in this work. Once the design strategy is finalized, it is necessary to have a proper implementation methodology. Since the operations on the input data, in the case of quadratic filters, are basically Kronecker products a direct implementation is

impractical. Matrix decomposition methods, especially the singular value decomposition is preferred in this work.

Once the design and implementation methodologies are selected, potential areas where the nonlinear effects due to polynomial products are predominant are identified as

- Edge detection
- Noise removal and
- Prediction of speech signals.

### 9.2.2 Edge Detection

Edges in images are high frequency components caused by peripheries of objects, the detection of which is a key image processing operation. Quadratic filters are used for detecting edges with improved performance than other nonlinear edge detection filters. The applications for which quadratic edge detection systems developed are:

- Unsharp masking scheme for enhancing latent fingerprints.
- Detection of retinal microaneurysms due to diabetic retinopathy

#### Unsharp masking for enhancing latent fingerprints

Fingerprint, being a unique biometric identifier, its enhancement and recognition are rich areas of image processing. It has applications in both access control and forensic sciences. The focus of work is on enhancing latent prints from crime scenes to ease forensic identification. The broken ridges and valleys in the print are enhanced by improving the contrast between dark ridges and light valleys. In the contrast enhancement scheme of unsharp masking, an image is added with the

scaled version of edges separated from it. If the images are noisy, as is often the cases with fingerprints from crime scenes, the performance of conventional edge detectors like Laplacian, LoG, Canny, Sobel etc. deteriorates as noise appears as false edges. Quadratic edge detection filters with greater noise invulnerability are proposed and implemented in the unsharp masking. It is observed to have better signal to noise ratio than other filters as shown in Fig. 9.1. Specifically, the SNR improvement in the case of quadratic filter is $\approx 4$ dB above the nearest competitor, LoG filter. The computational complexities of various filters, except that of Canny filter which is very high, are shown in Fig. 9.2.

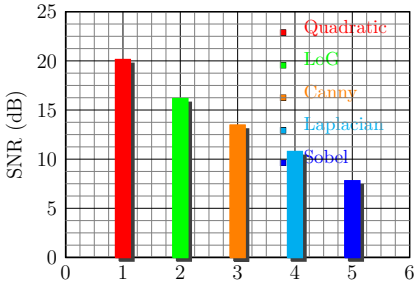



**Fig. 9.1.** Signal to noise ratio (dB) for various filters

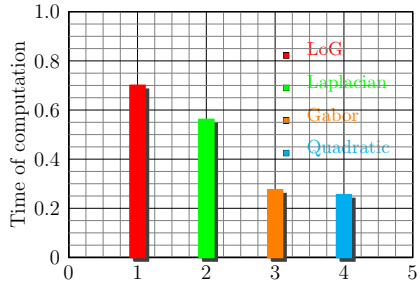



**Fig. 9.2.** Computational complexity for various filters

The quadratic filter using SVD method has the smallest complexity. The comparison of mean structural similarity index, a performance parameter for the preservation of structure of ridges, in presence of impulsive noise and Gaussian noise are in Fig. 9.3 and Fig. 9.4 respectively. Quadratic filter is observed to preserve the structure of ridges better than other filters in presence of both impulsive and Gaussian noise. LoG also performs almost at par with quadratic filter.

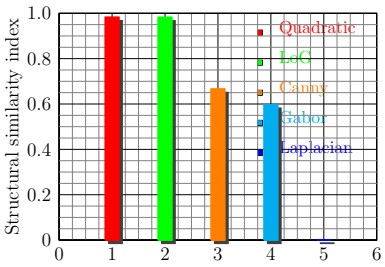



**Fig. 9.3.** SSIM for various filters under impulsive noise

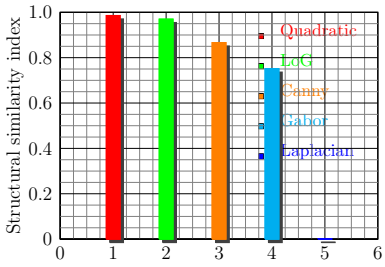



**Fig. 9.4.** SSIM for various filters under Gaussian noise

### Detection of Retinal Microaneurysms

Microaneurysms developed in the retina due to deposits of glucose and lipids caused by diabetes can silently worsen and result in permanent loss of vision. Surgery at an early stage is the cure. Automatic surgery is facilitated if the periphery of microaneurysm is enhanced from a fundus image. Three types of quadratic filters are designed and implemented for this end.

- Two dimensional Teager filter based on least square method.
- Two dimensional Teager filter based on minimization of mean square error.
- Two dimensional quadratic filter based on optimization.

The performance parameters in terms of the improvement in signal to noise ratio and the time of computation are compared with conventional edge detection filters as shown in Fig. 9.5. It is seen that the two Teager filters perform fairly identical, but better than conventional filters, although the complexity is more. The quadratic filter by optimization outperforms other filters with time of computation equal

to LoG filter, the nearest competitor, as shown in Fig. 9.6.

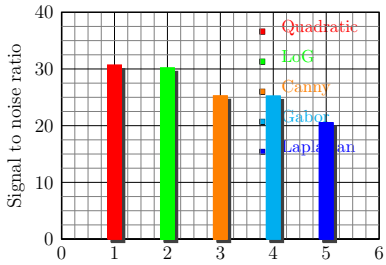



**Fig. 9.5.** Signal to noise ratio for various noise removal filters

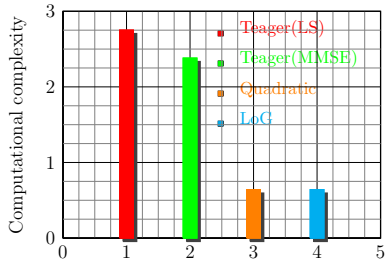



**Fig. 9.6.** Computational complexity of various filters

### 9.2.3 Noise Removal

Noise is any unwanted disturbance that gets added with the desired signal from outside the system or from within. They can also be due to the quantization of discrete signals during processing. Noise in communication systems is always Gaussian in statistics as a consequence of the central limit theorem, and well established methods like correlation detection, matched filtering etc. have been developed for separating the desired signal from the noise signal. But noise present in images are predominantly impulsive in nature, separation of which is difficult. Conventionally, nonlinear filters like mean and median filters are used for denoising images. Two dimensional quadratic filter designed based on the maximization of signal to noise ratio is proposed and implemented for removing impulsive noise from raw MRI data. The popular medical imaging scheme of MRI maps nuclear resonances, when subjected to strong magnetic fields, into intensity variations. The tumor tissue and body tissue have different resonant frequencies and this difference is used to contrast the tumour. The resolution of imaging is enhanced by changing the magnetic field rapidly.

These sharp changes introduces a great deal of impulsive noise into raw MRI data. The quadratic filter offered the highest signal to noise ratio of $\approx$ 10dB above that of minimum filter, as seen from Fig. 9.7.

In the case of MRI, the edges mean the periphery of a region of interest such as a tumour. Edges in images are usually corrupted by additive noise and most edge detectors are sensitive to noise. So the preservation of edges on filtering is very critical. Quadratic filters are proposed in this work as better edge detectors in terms of noise invulnerability and edge preservation. Fig. 9.8 shows the edge crispness function for various filters. A low value of crispness indicates better edge preservation. Quadratic filter preserves the edges so well that the sensitivity is too small to be visible in Fig. 9.8.

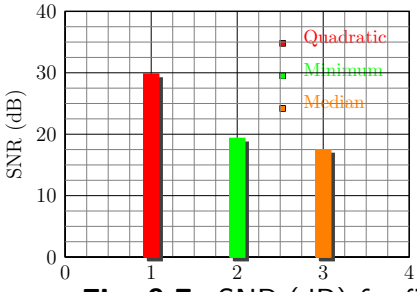



**Fig. 9.7.** SNR (dB) for filters

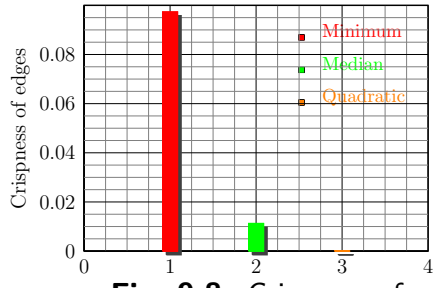



**Fig. 9.8.** Crispness of edges

### 9.2.4 Prediction of Speech Signals

The research consists of remodeling the differential pulse code modulation system for coding speech signals, by replacing the linear predictor in it by a quadratic predictor. The aim is to account for the polynomial components in speech signal. The modified system yields smaller mean square error between transmitted signal and received signal, on

passing through an additive white Gaussian noise channel, than that with linear predictor and quadratic lattice type predictor as shown in Fig. 9.9. It shows that the mean square error is roughly one third that of linear predictor, giving rise to better audio reception. The predictor based on optimization surpasses the quadratic lattice predictor in performance and in the ease of implementation.

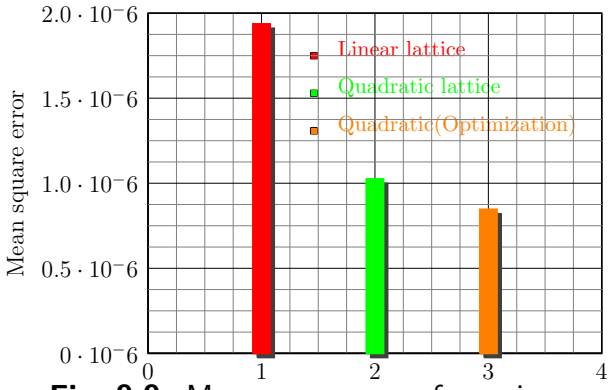



**Fig. 9.9.** Mean square error for various predictors

## 9.3 Limitations of the Study

The last chapters elucidated the applications and advantages of employing quadratic Volterra filters for edge detection, noise removal and statistical prediction. Use of quadratic Volterra filters resulted in improved performance parameters such as signal to noise ratio, edge crispness, mean square error etc. than conventional filters. However, there are noteworthy limitations that are to be circumvented by the designers. The major limitations in working with polynomial systems for discrete signal processing are summarized below.

### 9.3.1 Working without Frequency Domain

When working with LTI systems, one has the surety that only the input frequencies or a subset thereof can appear at the output. This is not the case with quadratic systems. There is no exact equivalent of frequency domain for quadratic systems. The familiar concepts with LTI systems such as the convenient input-output relationship, the spectral relationship and the much familiar computational tools like FFT are no longer applicable when working with quadratic or polynomial systems. Time domain filtering techniques need to be improved in their stead.

### 9.3.2 Higher Order Systems in Volterra Series

This research work is confined to quadratic systems with the assumption that majority of the effects due to polynomial nonlinearities are covered by the quadratic term in Volterra series expansion. This fails to exploit the effects due to cubic and higher terms, if there are any.

### 9.3.3 Difficulty in Hardware Realizations

Much of the DSP hardware in the market, based on Harvard architecture, are designed for LTI systems. They are not inherently designed for Kronecker products. This poses serious limitations when one tries to implement quadratic systems on DSP hardware.

## 9.4 Scope for Further Work

Polynomial systems, based on Volterra power series, is a relatively untrodden area of signal processing which offers a lot of challenges to researchers. In addition to the existing work, extensive research will be done with applications in image processing, computer vision,

communication etc. The limitations posed in Sec. 9.3 will be overcome by expansions of the present research work. It will be further extended in the following avenues.

### 9.4.1 Efficient Structures in Time Domain

Since the operations in the frequency domain are not easy with polynomial systems, it is imperative to look for efficient structures in the time domain. Matrix decomposition of the polynomial kernels coupled with distributed arithmetic will result in compact structures that can perform fast filtering.

### 9.4.2 Addition of Cubic Systems

Addition of cubic systems in parallel with quadratic systems can be done. This will be useful in edge detection as more high frequency components can be encompassed by the cubic term. The major challenge will be in harnessing the computational complexity arising from third order products.

### 9.4.3 Implementation on FPGA

The quadratic filters designed and tested for specific applications can be implemented on FPGA and can be used for real time applications. The parallel-cascade structure based on singular value decomposition is at par with conventional filters in terms of speed of execution. Further improvement in speed can be achieved by realizing each FIR filter in every channel based on distributed arithmetic.

The research work is summarized with emphasis on the key performance parameters of quadratic systems. They are contrasted with those of linear and conventional nonlinear systems to establish the

supremacy of quadratic systems for the applications of edge detection, noise removal and prediction. Also, the challenges in working with quadratic systems are presented. Possibilities of future expansions and further work are explored. Based on the inferences, impact of the work on various stakeholders is presented in the next chapter.

Chapter 10

# Research Contributions

## Contents



## 10.1 Overview

Digital signal processing, an engineering area that deals with the manipulation of discrete data using discrete time systems, revolutionized and enriched many fields like communication engineering, digital audio, data analysis, remote sensing, instrumentation etc. The theory and the systems matured over four decades, but much of the progress revolved around LTI systems, ignoring nonlinear behaviour in systems and nonlinear elements in signals grossly. The first objective of the

present research is to make a comparative study of linear and nonlinear systems. Real life systems show nonlinear characteristics and many signals are generated by nonlinear processes. The comparative study opened up possible avenues where effects of polynomial nonlinearities could be exploited. The second major objective is to identify the suitable power series to model the input-output relation for a polynomial system. The present work proposed the use of quadratic Volterra systems in the attempt to model mild polynomial nonlinearities. Volterra power series, a variant of Taylor series, is used to model polynomial systems as parallel extensions of linear systems. Then, the general strategy for design and implementation of quadratic systems are reached at. Focus of the work is on quadratic systems with applications in prediction of speech signal, edge detection and impulsive noise removal. These three areas are identified as potential avenues where quadratic systems could outperform conventional linear systems in terms of various performance parameters.

The work is carried out with the objective of modeling polynomial nonlinearities using quadratic filters with targeted applications in

- Prediction
- Edge detection and
- Noise removal

The methodology adopted consists of the study of quadratic systems, their design and realization for the specific application and testing for assessing performance parameters. The main challenges are the lack of general design rules and the computational complexity of implementation. The application specific design methods consist of optimization of a key objective function and consequent solution of the filter kernel. This method has the advantage that quadratic filters tailored for the

specific application could be designed. The computational complexity arising from direct implementation is surmounted by resorting to approximate structures based on matrix decomposition. The work is validated by comparing the performance with those of conventional systems under identical working environment.

Once the designed mathematical models are put into practical implementations, it is necessary to test them exhaustively. Experiments are designed to test the filters for edge detection, noise removal and prediction. The parameters are compared with those of LTI systems and other nonlinear systems. Once the validations are done, the filters are put to practical applications. The impacts of these practical applications to various stakeholders are detailed in the next section.

## 10.2 Research Contributions

Quadratic systems are developed for edge detection, noise removal and prediction, the practical applications of which include fingerprint enhancement, detection of retinal microaneurysms, MRI data denoising and quadratic prediction of speech signals. These works influence many, like researchers, doctors, police, forensic experts and communication engineers. The novelty introduced in the research work in the three application areas and their impacts are as outlined below.

### 10.2.1 Impacts to Researchers

Theory and development of polynomial systems in general, and quadratic systems in particular, are unforayed areas of research. The present work is an attempt to employ quadratic systems for nonlinear signal and image processing. With the completion of this work, more literature has been added to polynomial signal processing. Ten papers are

generated out of this research which throw light into a different kind of signal processing area. The challenges like design, computational complexity reduction and the absence of frequency domain tools make polynomial signal processing a promising area of research.

### 10.2.2 Impacts to Practitioners

Besides the impacts to researchers, the present research work will influence many practitioners also. They include medical practioners, surgeons, police, forensic experts and engineers.

#### Medical Practitioners

Ophthalmic surgeons will benefit from the work done on retinal images. The filters developed, if incorporated into a fundoscope, will enhance even very small microaneurysms due to diabetic retinopathy.

- Teager algorithm, for energy detection, and least squares method are used to design a two dimensional filter for detecting retinal microaneurysms.
- Two dimensional Teager filter based on MMSE method is designed and implemented for detecting retinal microaneurysms.
- A general quadratic filter based on optimization algorithm is designed and implemented for detecting retinal microaneurysms.

Since the exact detection of periphery of microaneurysms is possible, the work is useful in automated surgery.

Oncologists will benefit from the work done on the denoising of raw MRI signals using quadratic filters. This filter, if used to replace the conventional order statistic filters, will enhance the noisy MRI so that the peripheries of tumours will be more pronounced.

### Police and Forensics

In this work, a quadratic edge detection filter is designed and implemented and is incorporated it in unsharp masking scheme for enhancing latent fingerprints. The filter is tested against conventional edge detection filters on raw latent fingerprints. This method enhances the noisy ridges in fingerprints better than other filters and is observed to preserve the structure of the image even at high noise levels. This unsharp masking will be useful as a preprocessing stage for forensic experts before subjecting the crime scene fingerprint to the national database for the detection of the identity of criminals.

### Communication Engineers

Communication engineers benefited a lot from the research in signal processing. Most present day communication systems run on DSP algorithms. Polynomial systems will also have its share of contributions. In the research work, quadratic predictor based on the minimization of mean square error is designed, implemented and incorporated in a differential pulse code modulator and demodulator. The new system is observed to have better performance in terms of audio quality at the receiver and the mean square error between transmitted and received signal over an additive white Gaussian noise channel.

## 10.3 Conclusion

Recently, discrete Volterra series emerged as a potential candidate to represent polynomial characteristics in physical systems as implementations of quadratic, cubic and higher order systems in parallel with the conventional LTI systems. Such implementations rendered better performance values than those offered by LTI or other nonlinear filters. Quadratic systems, in particular, had gained popularity in signal

and image processing, despite the complexity in implementations. The present research work consists of design and development of quadratic filters for edge detection, noise removal and prediction. The filters for edge detection are put to two applications *viz.* the enhancement of latent fingerprints and the detection of retinal microaneurysms.

An unsharp masking scheme based on a quadratic edge detection filter is used to enhance latent fingerprints, obscured by noise. The new scheme is observed to preserve the ridge - valley structure better than schemes employing conventional edge detectors. The results numerically validate the enhanced visual quality of fingerprints by the quadratic method, which conventional methods could not achieve.

Two dimensional Teager filters based on least squares and minimum mean square error methods are designed and implemented for the detection and localization of retinal microaneurysms due to diabetic retinopathy. A quadratic edge detection filter, based on optimization, is designed and implemented for the same purpose. This filter enhanced the microaneurysms better than Teager filters even under strong impulsive noise. All the three methods are compared among themselves and with conventional edge detection methods by testing on fundus retinal images. The results establish the suitability of quadratic edge detection filter for the detection and localization of microaneurysms.

The work also proposes a quadratic de-noising filter for the removal of impulsive noise from raw MRI signals and is found successful. The performance is established superior to the nonlinear de-noising filters such as median and mean filters.

The quality of reception of speech signals in a differential pulse code modulation receiver is enhanced by the inclusion of a quadratic pre-

dictor at the transmitter and receiver. This modification is done to account for the polynomial terms that gets added during the generation of speech signal. The subjective quality on hearing and the improved mean square error between the transmitted and received signal are indicative of the supremacy of quadratic predictor over linear predictor.

The work establishes the superiority of quadratic filters over conventional methods in the tasks of edge detection, noise removal and prediction. The research forays into the unfamiliar vistas of polynomial signal processing with the motivation to look beyond LTI systems. The endeavor also adds to the volume of literature in quadratic filtering which, hopefully, will be useful for future researchers in the fertile field of polynomial signal processing. The work impacts practitioners in electronic communication, signal processing, image processing, ophthalmology, oncology and forensic science.

# List of Publications

1. Hari V S, Jagathy Raj V P and Gopikakumari R (2013) Teager and Quadratic Filter for the Enhancement of Edges in Retinal Images for the Efficient Detection and Localization of Diabetic Retinopathy(Communicated to *IEEE Trans. on Biomedicine.*)

2. Hari V S, Jagathy Raj V P and Gopikakumari R (2013)Performance Analysis of Differential Speech Coders based on Quadratic Predictors(Communicated to *International Review of Electrical Engineering.*)

3. Hari V S, Jagathy Raj V P and Gopikakumari R (2012) Enhancement of Fingerprints using Unsharp Masking based on Quadratic Edge Detection Filter, *Elsevier Journal of Pattern Recognition.* (Published)

4. Hari V S, Jagathy Raj V P and Gopikakumari R (2012).Enhancement of Calcifications in Mammograms using Volterra series based Quadratic Filter In *Proc. of the International Conference on Data Science and Engineering (ICDSE)*, 978-1-4673-2149-5/12/$ 31.00©2012 IEEE, July 118-20,2012, pp. -137–144

5. Hari V S, Jagathy Raj V P and Gopikakumari R (2012). Spatial Filtering of MRI Images using Quadratic Filter based on Volterra

Series. *International Journal on Communication, Antennas and Propagation(IRECAP)*, ISSNS 2039-5086, pp. 252–258.

6. Hari V S, Jagathy Raj V P and Gopikakumari R (2012). Quadratic Predictor based Differential Encoding and Decoding of Speech Signals In *Proc. of the International Conference in Communication and Signal processing*, 978-1-4244-9799-7/11/$ 26.00©2011 IEEE

7. Hari V S, Jagathy Raj V P and Gopikakumari R (2011). Quadratic Edge Detection Filter for the Enhancement of Fingerprints in Noisy Background In *Proc. Control Instrumentation System Conference (CISCON)*, pp. 474–477.

8. Hari V S, Jagathy Raj V P and Gopikakumari R (2011). Noise Smoothening of Retinal Images by Quadratic and Rational Filters In *Proc. Control Instrumentation System Conference (CISCON)*,, pp. 344–347.

9. Hari V S, Jagathy Raj V P and Gopikakumari R (2011) Python - A Novel Tool for Biomedical Imaging *Proc. of Workshop on Scientific Computing*

10. Hari V S, Jagathy Raj V P and Gopikakumari R (2010). Volterra Series based Polynomial Predictor for Speech Signals. In *Proc. of International Conference on Control, Communication and Computing.* pp. 300–303.

# Bibliography

Abuelmatti, M. H. (1990). Harmonic and intermodulation distortion of carbon microphones. *Applied Acoustics*, 31:233–243.

Adams (1991). FIR digital filters with least square stop bands subject to peak gain constraints. *IEEE Trans. on Circuits and Systems*, 39(4):376–388.

Adams, S. et al. (1993). New approaches to constrained optimization of digital filters. In *Proc. IEEE Int. Symp. Circuits and Systems (ISCAS)*, pages 80–83.

Alper, P. (1963). A consideration of the discrete volterra series. *IEEE Trans. on automatic control*, AC-8:322–327.

Arun Ross, A. J. and Reisman, J. (2003). A hybrid fingerprint matcher. *Elsevier J. Pattern Recognition*, (36):1661–1673.

Athanasios Papoulis, S. U. P. (2002). *Probability, random variables and stochastic processes*. Mc Graw Hill, New York.

Bellamy, J. C. (2000). *Digital Telephony*. Wiley, New York, 3rd edition.

Bergland, G. D. (1968). A fast fourier transform algorithm using base 8 iterations. *Math. Comp.*, 22:275–279.

Bloch, H. and Packard (1946). Nuclear infraction. *Phys. Rev.*, 69:127–137.

Bluestein, L. I. (1970). A linear filtering approach to the computation of discrete fourier transform. *IEEE Trans. Audio Electroacoustics*, AU-18:451–455.

Brent, R. P. (1973). *Algorithms for minimization without derivatives*. Englewood Cliffs, N.J.

Cooley, T. (1965). An algorithm for the machine calculation of fourier series. *Math. Comp.*, 19:297–301.

Damadian, Minkoff, S. and Field, K. (1976a). Focusing nuclear magnetic resonance (fonar): visualization of a tumor in a live animal. *Science*, 194:1430–32.

Damadian, Minkoff, S. and Field, K. (1976b). Tumor imaging in a live animal by focusing nmr (fonar). *Physiol. Chem. Phys.*, 8:61–65.

Dougherty, G. (2011). *Medical Image Processing: Techniques and Applications*. Springer, 1st edition.

Ege, B., Hejlessen, O. K., Larsen, O. V., et al. (2000). Screening for diabetic retinopathy using computer based image analysis and statistical classification. *Computer methods and programs in biomedicine*, (62):165–167.

Ekstrom, M. (1984). *Digital Image Processing Techniques*. Academic Press.

Eskicioglu, A. M. and Fisher, P. S. (1995). Image quality measures and their performance. *IEEE Trans. Communications*, 43:2959–2965.

Fletcher, R. and Powell, M. J. D. (1963). A rapidly convergent descent method for minimization. *Computer J.*, (6):163–168.

Gantmacher, F. R. (1960). *The Theory of Matrices*. Chelsea, New York.

Gardner, Keating, W. and Elliot (1996). Automatic detection of diabetic retinopathy using an artificial neural network:a screening tool. *Br. J. Ophthalmol.*, (80):940–944.

Geva, T. (2006). Magnetic resonance imaging: Historical perspective. *Cardiovascular Magnetic Resonance*, 8:573–580.

Girod, B. (1993). Whats wrong with mean-squared error? In Watson, A. B., editor, *Digital Images and Human Vision*, pages 207–220. MIT Press.

Gold, L. R. R. B. (1975). *Theory and Applications of Digital Signal Processing*. Prentice Hall.

Gonzalez, R. and Woods, R. (1992). *Digital Image Processing*. Addison-Wesley, Reading,MA, 3rd edition.

Hayes, M. H. (2003). *Statistical Digital Signal Processing and Modeling*. John Wiley, Singapore.

Heideman, J. D. and Burrus (1984). Gauss and the history of the fast fourier transform. *IEEE ASSP Magazine*, 1(4):14–21.

Hermann, O. (1970). Design of nonrecursive filters with linear phase. *Electron. Lett.*, 6(11):328–329.

Hopf, E. (1934). *Mathematical problems of radiative equilibrium*. Cambridge University Press.

Hou, Z. J. and Wei, G. W. (2002). A new approach to edge detection. *Elsevier J. Pattern Recognition*, (35):1559–1570.

Ilea, D. E. and Whelan, P. F. (2011). Image segmentation based on the integration of colourtexture descriptors - a review. *Elsevier J. Pattern Recognition*, (44):2479–2501.

Jain, A. K. (2003). *Fundamentals of Digital Image Processing*. PHI, New Delhi.

Jain, A. K. and Prabhakar, S. (1999). A multichannel approach to fingerprint classification. *IEEE Trans. on pattern analysis and machine intelligence*, 2(4):1559–1570.

Jan, J. (2005). *Medical Image Processing, Reconstruction and Restoration: Concepts and Methods*. CRC Press, 1st edition.

Kaiser, J. F. (1990). On a simple algorithm to calculate the energy of a signal. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 381–384.

Kaiser, S. (1983). Design of FIR filters with flatness constraint. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing(ICASSP)*, pages 197–200.

Koh, T. and Powers, E. J. (1985). Second-order volterra filtering and its application to nonlinear system identification. *IEEE Transactions on acoustics, speech, and signal processing*, assp-33(6):1445–1455.

Köse, C., Şevik, U., Larsen, O. V., et al. (2011). Simple methods for segmentation and measurement of diabetic retinopathy lesions in

retinal fundus images. *Comput. Methods Programs Biomed. Meas.*, (3244).

Lea, T. H. and Van, H. T. (2012). Fingerprint reference point detection for image retrieval based on symmetry and variation. *Elsevier J. Pattern Recognition*, (45):3360–3372.

Maltoni, D., Maio, D., Jain, A. K., and Prabhakar, S. (2003). *Handbook of Fingerprint Recognition*. Springer,New York.

Mathews, V. J. and Sicuranza, G. L. (2000). *Polynomial Signal Processing*. John Wiley and Sons Inc.,New York.

Mayer-Baese, U. (2007). *Digital signal Processing with Field Programmable Gate Arrays*. Springer, New York, 3rd edition.

McClellan, J. H. and Parks, T. W. (2005). A personal history of the parks-mcclellan algorithm. *IEEE Signal Processing Magazine*, 22(2):82–86.

Mendonca, C. and Nunes (1999). Automatic segmentation of microaneurysms in retinal angiograms of diabetic patients. In *Proc. IEEE Int. Conf. of Image Analysis Applications(ICIAP 99)*, pages 728–733.

Mitra, S. K. and Sicuranza, G. (2001). *Nonlinear Image Processing*. Academic Press Series in Communication, Networking and Multimedia, San Diego.

Mix, D. F. (1995). *Random Signal Processing*. Prentice Hall, New York.

Noble, B. (1988). *Methods based on the Wiener-Hopf Technique*. Chelsea Publishing Company, New York, 2nd edition.

Nowak, R. D. and Veen, B. D. V. (1994). Random and pseudorandom inputs for volterra filter identification. *IEEE Trans. Signal Processing*, 42(8):2124–2135.

Nyquist, H. (2002). Certain topics in telegraph transmission theory. *Proc. IEEE*, 90(2):280–305.

Øien, O. (1995). Diabetic retinopathy: automatic detection of early symptoms from retinal images. In *Proc. Norwegian Signal Processing Symposium*, pages 135–140.

Oppenheim, A. V. and Schafer, R. W. (1998). *Discrete Time Signal Processing*. PHI, New Delhi, 5$^{\text{th}}$ edition.

Pappas, T. N. and Safranek, R. J. (2000). Perceptual criteria for image quality evaluation. In Bovik, A., editor, *Handbook of Image and Video Processing*. Academic Press.

Pratt, W. K. (2001). *Digital Image Processing*. John Wiley and Sons Inc.,New York.

Proakis, J. G. (2001). *Digital Communications*. Mc Graw Hill, New York.

Proakis, J. G. and Manolakis, D. A. (1998). *Digital Signal Processing*. PHI, New Delhi, 5$^{\text{th}}$ edition.

Purcell, T. and Pound (1946). Resonance absorption by nuclear magnetic moments in a solid. *Phys. Rev.*, 69:37–38.

Rabi, Zacharias, M. and Kusch (1938). The molecular beam resonance method of measuring nuclear magnetic moments. *Phys. Rev.*, 55:526–535.

Rabiner, M. and Parks (1967). FIR digital filter design using weighted Chebyshev approximation. *Proc. IEEE.*, 55:149–171.

Rader and Gold (1975). Digital filter design techniques in the frequency domain. *Proc. IEEE.*, pages 595–610.

Ramponi, G. (1986). Edge extraction by a class of second-order nonlinear filters. *Electon Lett.*, 22(9):482–484.

Ramponi, G. (1990). Bi-impulse response design of isotropic quadratic filters. *Proc. of the IEEE*, 78(4):665–677.

Ramponi, G. and Ukowich, W. (1987). Quadratic 2-d filter design by optimization techniques. In *Proc. Int. Conf. Digital Signal Processing*, pages 59–63.

Roberts, R. A. and Mullis, C. T. (1987). *Digital Signal Processing.* Addison-Wesley, 1st edition.

Scheibert, P. and Debregeas (2009). The role of fingerprints in the coding of tactile information probed with a biomimetic sensor. *Science*, 323(5920):1559–1570.

Shifeng Lia, Huchuan Lua, A. J. and Zhang, L. (2012). Arbitrary body segmentation in static images. *Elsevier J. Pattern Recognition*, 45:3402–3413.

Sicuranza, G. L. (1992). Quadratic filters for signal processing. *Proc. of IEEE*, 80(8):1263–1285.

Singleton, R. C. (1967). A method for computing the fast fourier transform with auxiliary memory and and limit high speed storage. *IEEE Trans. Audio Electroacoustics*, AU-15:91–98.

Singleton, R. C. (1969). An algorithm for computing the mixed radix fast fourier transform. *IEEE Trans. Audio Electroacoustics*, AU-17:93–103.

Sleightholme, A. et al. (1994). Computer-aided digitization of fundus photographs. *Clin. Phys. Physiol. Meas.*, (5):295–301.

Spencer, P. et al. (1991). Automated detection and quantification of microaneurysms in fluroscein angiograms. *Graefe's Arch. Clin. Exp. Ophthalmol.*, (23):46–53.

Struik, D. J. (1969). *A Source Book in Mathematics 1200-1800*. Harward University Press, Cambridge, Massachusetts.

Thurnhofer, S. and Mitra, S. K. (1996). A general framework for quadratic volterra filters for edge enhancement. *IEEE Transactions on image processing*, 5(6):950–963.

Vizireanu, N. and Udrea, R. (2007). Iterative generalization of morphological skeleton. *Journal of Electronic Imaging*, 16(1):1–3.

Vizireanu, N. and Udrea, R. (2009). Visual-oriented morphological foreground content grayscale frames interpola tion method. *Journal of Electronic Imaging*, 18(2):1–3.

Walker, G. (1931). On periodicity in series of related terms. *Mon. Wea. Review.*, (59):277–278.

Wang, Alan C Bovik, H. R. S. and Simoncelli, E. P. (2004). Image quality assessment: from error visibilty to structutal similarity. *IEEE Trans. on Image Processing*, 13(4):1–14.

Wang, Z. and Bovik, A. C. (2002). A universal image quality index. *IEEE Signal Processing Lett.*, 9:81–84.

Weeks, A. R. (2005). *Fundamentals of Electronic Image Processing*. Prentice Hall India.

White, S. A. (1989). Applications of distributed arithmetic to digital signal processing: A tutorial review. *IEEE ASSP Magazine*, 6:4–19.

Wiener, N. (1942). Response of a nonlinear device to noise. *Report No. 129, Radiation Laboratory, MIT, Cambridge, Massachussets*.

Wiener, N. (1956). *I am a Mathematician*. Doubleday & Co., Garden city, New York.

Winograd, S. (1976). On computing the discrete fourier transform. *Proc. Natl. Acad. Sci.*, 73:105–106.

Xu, W. and Hauske, G. (1994). Picture quality evaluation based on error segmentation. *Proc. SPIE*, 2308:1454–1465.

Yule, G. (1927). On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. *Phil. Trans. Roy. Soc. London*, (A226):267–298.

Appendix A

# Python - A Computational Tool for Biomedical Signal Processing

## A.1 Introduction

Scientific computing is a key area of research which involves in developing platforms that cater the requirements of scientific community. It often requires modeling of complex systems, simulation of scientific processes, data mining, statistical interpretation of data etc. Large arrays of data need be processed with minimum programming difficulty in short time. Although there are many high level programming languages like C, C++ etc that are suitable for general computing, there are but a few that are usable for scientific computing purposes. The major requirements for a language for scientific computing are

- The language need be interpreted and not compiled. The line by line interpretation can easily return results, a feature which speeds up research.
- The language should have a simple syntax so that less time is spent on programming and more time is spent on scientific thinking.

- The language should support a high quality plotting library that can generate publication quality plots, preferably in Postscript format.
- There need be easy syntax for easy vector- matrix multiplication, linear algebra, Fast Fourier transforms etc.
- Greater execution speed.

Historically, FORTRAN had many features of a scientific computing language, though it did not have its own plotting library. C remained as a fast language for general computing. Being compiler based it did not become popular in scientific computing. As C did not have easy array multiplication syntax, MATLAB, a meta language, that supported a lot of good features came to picture with an immediate success. MATLAB has many disadvantages besides the extremely high cost and licensing hassles. Being a closed source program one can never ascertain how the computations are done. It does all computations with arbitrary precision. Although MATLAB has a plotting library, it does not provide truly publication quality plots. So there was a need to look around for alternatives. With developments in open source software tools like Octave, Scilab, R etc came to picture the details of which are discussed in the next section.

## A.2 Open source Tools for Scientific Computing

With the arrival of Linux and internet, distributed computing became popular and open source software developers around the globe could work as a team, resulting in improved software tools. In scientific computing clones of MATLAB like SCILAB, Octave came to picture. SCILAB relied on TCL/TK. Although it contains many toolboxes for

signal processing, image processing, communication, control systems etc. it lacked robustness. Furthermore the TCL/TK GUI is still not publication quality. Despite these facts it remains popular especially in academic circles. Octave has the same syntax as MATLAB but is a very slow in execution. There has been demand for an interpreted and modular language for scientific computing with a good plotting library. And Python entered the arena at this juncture.
Python is a modular, interpreted programming language with many resemblances with LISP and FORTRAN. Being modular, programmers could integrate the existing FORTRAN libraries for linear algebra such as LAPACK(Linear Algebra Package) and ATLAS(Automatically Tuned Linear Algebra package) with Python making an open source scientific computing package called Scipy. The details of Python and scipy are in the next section.

## A.3 Python, Scipy and Matplotlib

Python language that came into existence just half a decade before, has very desirable features of a scientific computing language, the major ones being modularity, speed and robustness. An enhanced Python shell called *ipython* can be installed along with Python. Ipython resembles the command line in MATLAB and IDL but supports more features like shell scripting functions. The package *scipy* can be imported into Python for scientific computing. A Python library called MATPLOTLIB, popularly called *pylab* was developed for computing and for generating anti-aliased, publication quality plots. The two modules *scipy* and *pylab* forms the backbone of scientific computing with Python. These modules can be imported with the commands

```
from scipy import *
from pylab import *
```

The two commands import all the modules in scipy and pylab. Selective importing can be done to keep the code light and robust as

```
from scipy import math
```

Pylab offers good quality GUIs in addition to publication quality plots. Fig.A.1 shows random ellipses plotted with pylab. Python together with scipy and pylab has become a stable computing platform especially for signal and image processing. They offer greater speed and precision than MATLAB or IDL.

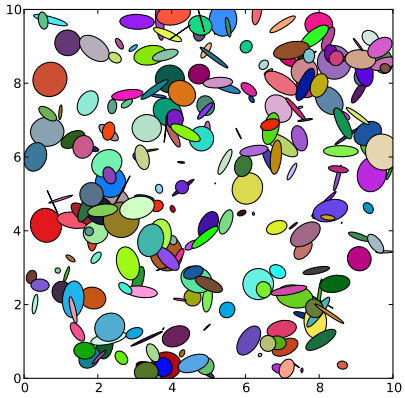



**Fig. A.1.** Random ellipses plotted in pylab

There are two main Python modules for image processing. The first one is the Python imaging library(PIL) and the second is a package in scipy called *ndimage*. Both contain modules for image acquisition, processing and image writing. Medical image processing comprises of reading, filtering, classifying, segmenting and interpreting medical images such as CT image, MRI image, retinal fundus image, angiograms

etc. Understanding the data in these images are very critical in both detection of diseases as well as in surgery and leaves no room for errors. Python and its image processing tools can facilitate medical imaging as detailed in the subsequent sections.

## A.4 Python for medical imaging

Both *ndimage* and *PIL* can be used with minimum amount of coding for processing medical images as both support variety of image formats like *.jpg*, *.png* , *.pgm* etc. The following code segment reads the color fundus image "retina.jpg"from the hard disk, isolates the green channel and displays it the pylab GUI as shown in Fig. A.2.

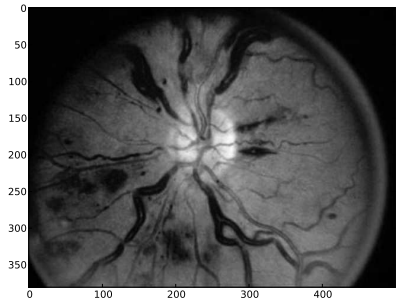



**Fig. A.2.** Fundus image with acute diabetic retinopathy

```
from scipy import*
from pylab import*
import Image
inpu=imread("retina.jpg")
x1=concatenate(concatenate(inpu))
x2=x1[1:len(x1):3]
```

```
x3=resize(x2,(shape(inpu)[0],
shape(inpu)[1]))
imshow(x3)
gray()
```

The image can be filtered for locating microaneurysms and swellings by a variety of filters based on scipy. Fig. A.3 indicates the result of median and Gaussian filtering as done the following code.

```
import scipy.ndimage as nd
subplot(121)
lapout=nd.gaussian_filter(x3,3)
imshow(lapout)
gray()
subplot(122)
medout=nd.median_filter(x3,3)
imshow(medout)
gray()
```

The histogram can be observed with the code given below. The bin size can be incorporated with an argument to the *hist* command as, say, *hist(x,100)*. The *grid* command includes a grid on the GUI. The *savefig* command saves the matplot lib GUI of the histogram in the present working directory. The figure can be saved in .png, .pdf or. jpg in addition to the encapsulated postscript(.eps) format.

```
hist(x3)
grid("True")
savefig("hist.eps")
```

The resulting histogram is as in Fig. A.4.

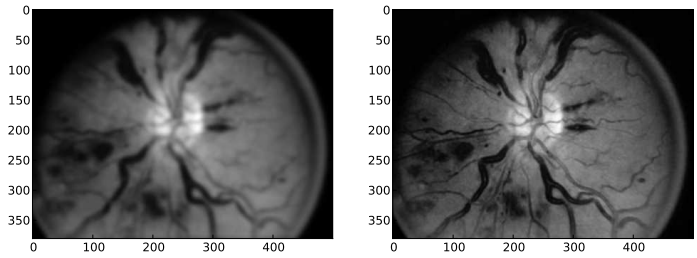



**Fig. A.3.** Gaussian filtered output(left) and median filtered output(right)

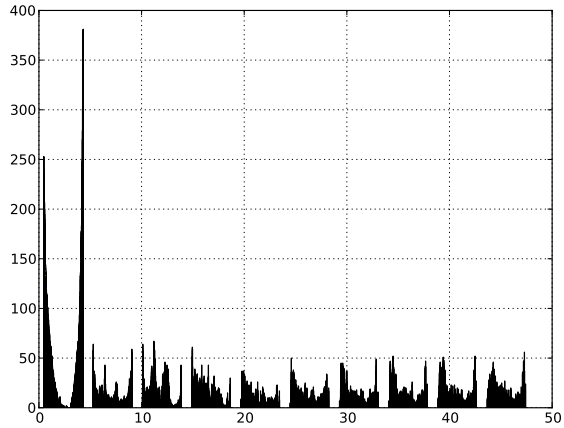
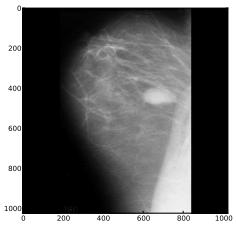


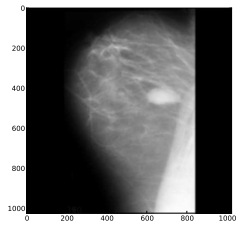

**Fig. A.4.** Histogram of retinal image

### A.4.1 Python for mammograms

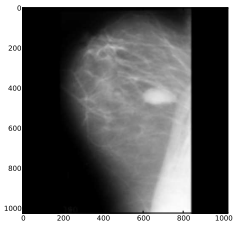
Mammograms in *.pgm* and other formats can be read and processed as discussed in the previous section. Fig. A.5 shows mammogram processed by various filters.



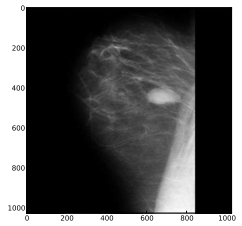
(a) Input mammogram



(b) Output of Sobel filter



(c) Output of LoG filter



(d) Output of quadratic filter

**Fig. A.5.** Outputs of various filters for mammogram input

## A.4.2 processing dicom files

Dicom files can be read and processed by the python module *pydicom*. The module can be imported as

```
import pydicom
```

Dicom is a format that is a lot less esoteric that one may imagine. Much of it contains the patient's information. The pathological data is contained as a $64 \times 64$ matrix which can be processed with scipy and pylab modules.

## A.5 Working with Sound Files

### *Wave* Module

The built *wave* module can be used for reading and writing sound waves in .wav, .raw etc. formats. The read waveforms can be easily converted into scipy arrays for further manipulations.

### *Wavfile* Module

It is a more light and robust module than *wave*.Sound files can be read using the *wavfile* in scipy.io module. It supports both the read and write functions. A sound file in .wav format can be read as

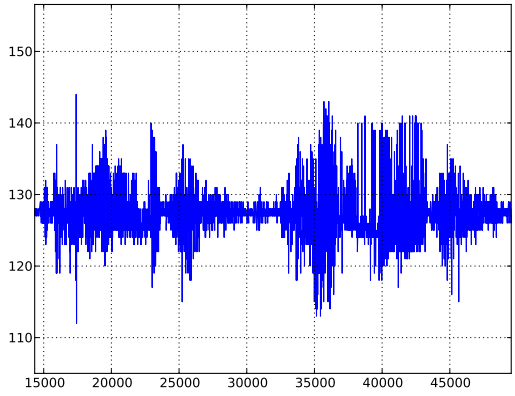



**Fig. A.6.** Signal from a wheezing lung

```
import scipy.io.wavfile as wav
z=wav.read("wheezinglungs.wav")
print z[0]
plot(z[1])
```

reads and displays the lung sound stored in the hard disk. The first element in the array z returns the sampling rate and the second array returns the values in the signal. The signal is as in Fig. A.6. The signal can be further processed using signal processing tools for feature extraction.

## A.6 Matplotlib Graphical User Interface

Matplotlib is Python library for publication quality outputs. This is invoked by importing the *pylab* module. This contains the *plot* function for continuous plots, *stem* function for discrete plots, *hist* for histograms etc. The plots are anti-aliased with LaTeX fonts for axes and text within the plot. Animations of plot are possible with pylab. the outputs can be stored in postscript, encapsulated postscript and in pdf format. This makes integration of results with reports compiled with LaTeX easily possible.

## A.7 Mayavi – Python Tool for Data Visualization

Mayavi is popular open source tool for scientific data visualization and 3-D plotting written in Python, currently hosted by *enthought*. It can be used as a stand alone application or as module that can be imported into a Python code. As a stand alone program it reads data in *vtk* format and displays in three dimensions. This make easy visualization of internal organs possible. Mayavi is used in 3-D visualization of MRI data. For using *mayavi* the Python session should be invoked as

```
ipython -wthread
```

Mayavi as a module can be imported as

```
import enthought.mayavi
```

Additional modules like scipy, pylab etc. can be imported and used along with mayavi.

## A.8 Installation of Various Modules

All the major Python modules can be installed using the *yum* or *apt* utility in Linux. The .exe files can be freely downloaded and installed on Windows machines.

## A.9 Conclusion

The chapter outlines the usage of Python as a stable replacement of commercial scientific computing languages like MATLAB, IDL etc. Scipy is a collection of signal processing and linear algebra routines written in FORTRAN that can be imported into Python for performing scientific computations quickly and reliably. MATPLOTLIB alias pylab offers publication quality plotting library. Modules like pydicom, ndimage etc. can be used for medical imaging. *Wavfile*, *wave* can be used for processing medical audio files. Medical data visualization can be done effectively using *Mayavi*.

# Index

## About the Author

Hari V S received the B Tech. degree in Electronics and Communication from University of Kerala in 1994 and the M Tech. degree in Electrical Engineering with specialization in Communication Systems from the Department of Electrical Engineering, Indian Institute of Technology, Madras in 2006. He spent 17 years in teaching in various engineering colleges under the Institute of Human Resources Development (IHRD) and is currently Associate Professor in charge of Principal, College of Engineering, Karunagappally.

He carried out his research work leading to PhD at School of Engineering, Cochin University of Science and Technology in Quadratic Filters based on Volterra series. The areas of interest are signal processing, polynomial systems and digital communication. The author supports open source software and has a passion for programming in Python for scientific computing and in promoting LaTeX for documentation. On a lighter side, he likes literature and uses it as an antidote for work.

Address
Punnamoottil, Manappally P O
Karunagappally, Kollam
Kerala, India – 690 574
Ph.- 91-8547463422
E-mail - hari_cec@yahoo.com