A COMPARISON BETWEEN PREVIOUSLY KNOWN AND TWO NOVEL IMAGE RESTORATION ALGORITHMS

Ali Said Ali Awad

M.S. Dissertation

Department of Electrical and Electronic Engineering
Eastern Mediterranean University

June, 2001
Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Zeka MAZHAR
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Electrical and Electronics Engineering.

Assoc. Prof. Dr. D. Z. DENİZ
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in cope and quality, as a thesis for the degree of Master of Science in Electrical and Electronics Engineering.

Assist. Prof. Dr. Erhan A. İNCE
Supervisor

Members of the examining committee
1. Assist. Prof. Dr. Hüseyin Bilgekul
2. Assist. Prof. Dr. Erhan A. İnce
3. Assist. Prof. Dr. Hasan Demirel
ABSTRACT

A COMPARISON BETWEEN PREVIOUSLY KNOWN AND TWO NOVEL IMAGE RESTORATION ALGORITHMS

Ali Said Ali Awad
M.S. in Electrical and Electronic Engineering
Eastern Mediterranean University
June 2001

Supervisor: Assist. Prof. Dr. Erhan A. İnce

Keywords: Wiener filter, iterative parallel data detection, 2D-VA, 2D VA-DF

Surveying the literature available indicates that a good amount of effort has been spent in the past decade trying to reconstruct bi-level images that have been degraded either by Additive White Gaussian Noise (AWGN) or by Inter Symbol Interference (ISI). Previous attempts concentrate on linear filtering techniques such as Inverse, Wiener, and Kalman type filtering. Such de-convolution methods have been found to be non-optimal. Other non-linear methods such as 1-D VA, 2D-VA, and 2D-VA with Decision Feedback are either sub-optimal or optimal, however the optimal 2D-VA suffers from computational complexity. Recently, Neifeld and Chugg have proposed a sub-optimal method called the Iterative Parallel Detection method. This method is fast and has less computational complexity due to its parallel architecture. This thesis presents the previous methods, as well as proposes two new sub-optimal algorithms. The first novel method is a sub-optimal algorithm based on linear filtering techniques and uses two Wiener filters and two thresholds. The second novel method is based on the Minimum Mean Square Error (MMSE) and a fixed threshold value and is quite successful in clearing images that have been degraded by AWGN and various types of blurs. Second new method is used for de-blurring images exposed to 1D type point spread functions. The simulation results are compared with those of other well established methods based on bit-error-rates and reconstructed versions of bi-level text-images.
DEDICATION

To my mother
To my father
To my wife Inas, my children Ameer and Amjd
To my cousin Mohammed
To my sisters
   Fatma
   Nama
   Naema
   Iman
And to my brothers
   Faize
   Ahmad
   Whaleed
   Faraje
   Khaled
Mohammed
Muneer
Abdallah
ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Assist. Prof. Dr. Erhan İnce for his continuous support and help throughout my research. Without his invaluable help I would not have been able to publish four-conference papers, two international and two national. He proved that he is more than patient for my enormous questions.

I would also like to thank all the staff members in the Electrical Department for their support and encouragement during my Master study.

Also I would like to thank my friend Ebrahim Soujeri who has also helped me a great deal, not only in my study but also to overcome the difficulties I have faced during my life in Cyprus.

Special thanks to my friends Loay Al-dappagh and Riad Amro from Mechanical and Civil Engineering respectively who has encouraged me to the end. Also I would like to thank my friends Turgay Çelik, Ali Hakan, Ahmet Rizaner, Zeer Alshinbari, Bassem Shomar and I am indebted to all the people who helped me during all my life.

My special thanks to my parents for their boundless encouragement and unlimited love. Whenever I do something, I always try to do the best for them.

I would like to thank my brothers particularly Faize and Ahmed, and also Fatma, my sister, and her husband Ramzi for their support and love all through my life.

A great thanks to my wife and children Ameer and Amjd who spend most of my Master Degree period away from me. I am forever indebted to them for their patience and unconditional love.

I will, of course, not forget to thank my wife's family, especially her father Dr. Ali Qaud who gave me the opportunity to continue with a Masters degree. Also, my thanks to her mother, brothers and sisters.

Finally, I would like to pay a special tribute to my cousin Mohammed Ali Awad who I didn't see for 29 years.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iv</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>v</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF SYMBOLS</td>
<td>x</td>
</tr>
</tbody>
</table>

## CHAPTERS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. DEGRADATION MODEL USED FOR IMAGE RESTORATIONS</td>
<td>5</td>
</tr>
<tr>
<td>2.1 General Degradation Model</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Degradation Model for Continuous Functions</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2 Degradation Model for Discrete Functions</td>
<td>7</td>
</tr>
<tr>
<td>3. INVERSE AND WIENER FILTERING TECHNIQUES</td>
<td>9</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>9</td>
</tr>
<tr>
<td>3.2 Inverse Filter</td>
<td>10</td>
</tr>
<tr>
<td>3.3 Wiener Filter</td>
<td>13</td>
</tr>
<tr>
<td>4. ITERATIVE DATA DETECTION TECHNIQUE</td>
<td>15</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>4.2 Page Oriented Optical Memory (POOM)</td>
<td>15</td>
</tr>
<tr>
<td>4.2.1 Algorithm Description</td>
<td>16</td>
</tr>
<tr>
<td>4.3 Double Threshold Based Novel De-blurring Technique</td>
<td>19</td>
</tr>
<tr>
<td>4.3.1 New Algorithm Description</td>
<td>19</td>
</tr>
</tbody>
</table>
5. BILEVEL IMAGE RESTORATION VIA 2D VA ..........................21
   5.1 Introduction.................................................................21
   5.2 Problem Statement.......................................................22
   5.3 Finding the Best Path in the Trellis.................................24
   5.4 Extension of the 1D- VA to 2D Problems............................26
   5.5 Viterbi Detection with Decision Feedback..........................28

6. NOVEL ONE OR TWO DIMENSIONAL DE-BLURRING ALGORITHMS
   BASED ON MINIMUM MEAN SQUARED ERROR AND A SELECTED
   FIXED THRESHOLD ..........................................................30
   6.1 Markov Process Modeling................................................30
   6.2 One Dimensional Novel De-Blurring Algorithm......................32

7. SIMULATION RESULTS......................................................38
   7.1 Image Restoration Using 1D Wiener and Inverse Filters..............38
   7.2 Image Restoration Using 2D Wiener and Inverse Filters..............42
   7.3 Novel Technique Based on Two Wiener Filters and Two Threshold Values....47
   7.4 Viterbi Algorithm Used for De-blurring Degraded Images.............50
   7.5 Novel One Dimensional De-Blurring Algorithm Based on Minimum Mean
       Squared Error and A Selected Fixed Threshold.........................52

8. CONCLUSION AND FUTURE WORK.........................................55
   8.1 Conclusion.................................................................55
   8.2 Future Work...............................................................56

REFERENCES.................................................................58
LIST OF FIGURES

Fig 2.1. Image degradation model.
Fig 3.1. Schematic diagram of a filter emphasizing its role in reshaping the input to match a desired signal.
Fig 3.2. Blurring a unit impulse function to obtain $H(u,v)$.
Fig 3.3. Image restoration by using Inverse filter at different values of $H(u,v)$.
Fig 4.1. Block diagram representation for Iterative Data Detection.
Fig 4.2. Double threshold -based architecture.
Fig 5.1. Four state trellis diagram.
Fig 5.2. 2D degradation model.
Fig 5.3. State definition for a system with $(3 \times 3)$ transfer function corresponds to the VA detector.
Fig 5.4. State definition for a system with $(3 \times 3)$ transfer function corresponds to the VA-DF detector.
Fig 6.1. The transmitted data in the original space $b^{3 \times 20}$ and the observation space $\mathfrak{R}^{1 \times 20}$ for a $3 \times 3$ PSF $\mathfrak{R}^{3 \times 3}$
Fig 6.2. Forming states and transitions from two dimensional data.
Fig 6.3. Partitioning 1D and 2D data into smaller stripes or areas.
Fig 6.4. Mean squared error computations and thresholding.
Fig 6.5. Matching number of states in different stages of the trellis.
Fig 6.6. Frame processing.
Fig 6.7. Trellis for a sequence of 10 input bits.
Fig 7.1. Comparison of BER performance for Wiener and Inverse filters at $K$-values of 0.1, 0.01, and 0.001.
Fig 7.2. Text-image restoration using Wiener and Inverse filters at $18$dB SNR.
Fig 7.3. Parametric Wiener and Inverse filters at $\Upsilon=1, 0.1, 0.01$.
Fig 7.4. Performance comparison of Wiener filter and Inverse filter for $h(0,j) = \lfloor 1/3 \rfloor$, $j=0,1,2$ and $h(0,j)=\lfloor 1/5 \rfloor$, $j=0,1,\ldots,4$. 
Fig 7.5. Performance comparison with lowpass masks $h_1$ and $h_2$.

Fig 7.6. Text-image reconstructions for images blurred by lowpass masks $h_1$ and $h_2$.

Fig 7.7. Performance curves for highpass filtering using $h_1$, $h_2$ and $h_3$.

Fig 7.8. Text-image restorations for images blurred by highpass masks.

Fig 7.9. Image reconstruction by using WF and IF.

Fig 7.10. BER performances for vertical blur $h_1$.

Fig 7.11. BER performances for origin symmetric blur $h_2$.

Fig 7.12. BER performances for Gaussian blur $h_3$.

Fig 7.13. Restoring an image corrupted by Gaussian blur $h_2$.

Fig 7.14. Performance for origin symmetric PSF $h_1$.

Fig 7.15. Performance for origin symmetric worst-case PSF.

Fig 7.16. Text restoration at 12dB for origin symmetric mask.

Fig 7.17. Performance for a Gaussian type mask.

Fig 7.18. Novel trellis method compared to other techniques.

Fig 7.19. Performance comparisons between new trellis method and conventional hard & soft detection VA.
# LIST OF SYMBOLS and ABBREVIATIONS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⊗</td>
<td>Convolution</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>Noise Variance</td>
</tr>
<tr>
<td>T</td>
<td>Transpose</td>
</tr>
<tr>
<td>P</td>
<td>Page</td>
</tr>
<tr>
<td>$E$</td>
<td>Average</td>
</tr>
<tr>
<td>$\ast$</td>
<td>Conjugation</td>
</tr>
<tr>
<td>o/p</td>
<td>Output</td>
</tr>
<tr>
<td>i/p</td>
<td>Input</td>
</tr>
<tr>
<td>$r(i,j)$</td>
<td>Received Elements</td>
</tr>
<tr>
<td>$h$</td>
<td>Impulse Response or Blur Matrix</td>
</tr>
<tr>
<td>$\hat{f}(x,y)$</td>
<td>Estimated Image</td>
</tr>
<tr>
<td>$f(x,y)$</td>
<td>Original Image</td>
</tr>
<tr>
<td>$g(x,y)$</td>
<td>Degraded Image</td>
</tr>
<tr>
<td>$\Pi$</td>
<td>Product</td>
</tr>
<tr>
<td>2D</td>
<td>Two Dimensional</td>
</tr>
<tr>
<td>1D</td>
<td>One Dimensional</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Zero Vector</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Lagrange Multiplier</td>
</tr>
<tr>
<td>AWGN</td>
<td>Additive White Gaussian Noise</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>FFT</td>
<td>Fast Fourier Transform</td>
</tr>
<tr>
<td>IF</td>
<td>Inverse Filter</td>
</tr>
<tr>
<td>ISI</td>
<td>Inter Symbol Interference</td>
</tr>
<tr>
<td>IFFT</td>
<td>Inverse Fast Fourier Transform</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A posteriori Probability</td>
</tr>
<tr>
<td>$Min$</td>
<td>Minimum</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>POOM</td>
<td>Page Oriented Optical Memory</td>
</tr>
<tr>
<td>PSF</td>
<td>Point Spread Function</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>TH, Th</td>
<td>Threshold</td>
</tr>
<tr>
<td>VA</td>
<td>Viterbi Algorithm</td>
</tr>
<tr>
<td>VA-DF</td>
<td>Viterbi Algorithm with Decision Feedback</td>
</tr>
<tr>
<td>WF</td>
<td>Wiener Filter</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In today’s world most document delivery services as well as various digitization activities are based on processing of bi-level images (black-and-white images). These digital images can provide access to most important textual and partly graphical information contained in newspapers, journals, and other types of printed documents and modern records. The bi-level image means that for each pixel (picture element) only one bit is enough to represent it (that is 1/24 part of the pixel size from the true color image). Generally one-bit pixel can express only black or white color, but if the number of pixels per unit area (image resolution) is sufficiently higher, such a solution can bring satisfactory results. Many systems in widespread use concentrate on the imaging of binary objects: i.e. the archival storage of text documents on microfilm, the facsimile transmission of text [1]. The readback signals are generally distorted due to the imperfect fidelity of such systems, the dispersion of the media as well as the write-and readback-channel neighbouring bits that affect each other and cause amplitude distortions and peak shifts [2]. Blur and Additive White Gaussian Noise will unavoidably distort these images. The blur could be due to a one-dimensional or a two-dimensional point spread function (PSF) and depends also on the density of pixels.

The question is how to restore the corrupted image by estimating a new image that has the same characteristics and features as the original one. The field of image restoration (sometimes referred to as image deblurring) is concerned with the estimation of the uncorrupted image from a distorted and noisy copy. Essentially, it tries to perform an operation on the image, which is the inverse of the imperfection in the image formation system. While simulating the use of image restoration methods, the characteristics of the degraded system and the noise can be assumed as known. In practical situations however one usually has hardly enough knowledge in order to obtain this information directly from the image formation process. The goal of image identification is to estimate the properties of the imperfect imaging system from the observed degraded image itself prior to the restoration process [4]. Approaching the
image restoration problem presents several other choices as well. First, the development can be done using either continuous or discrete mathematics. Second, the development can be carried out in either the spatial or frequency domain. Finally, while the implementation must be done digitally, the restoration can be effective in either the spatial domain (e.g., via convolution) or the frequency domain (via multiplication) [5]. Linear, position invariant processes can approximate most degradations. The linearity property makes the extensive tools of linear system theory become available for the solution of image restoration problems. Since the introduction of digital image restoration in the sixties, a variety of image restoration methods have been developed. Nearly all these methods assume that the point-spread-function (blur or transfer function of the imaging system) of the image formation system is known, and is therefore called a priori restoration.

One of the first methods used in image restoration was to simply neglect the presence of noise and to invert the blur through a frequency domain approach (Inverse filter) [6]. Since the signal spectrum normally dies out with frequency faster than that of the noise, the high frequencies are often dominated by noise. Also due to the magnitude of the Inverse filter increases with frequency, the filter enhances high frequency noise. Helstrom in [7], adopted the minimum mean square error estimation and presented the Wiener deconvolution filter, which afford an optimal method for rolling off the deconvolution point spread function in the presence of noise. Another solution to linear squared error image restoration uses a Kalman filter [8]. Linear filters based methods such as Inverse filtering, Wiener filtering, and Kalman filtering would result in poor performance improvements. It has been stated in [2,9] that the luck of extensive improvement in BER performances were due to the fact that a-priori information hidden in the original image is not utilized by such methods. A more effective a pproach is to use a non-linear technique such as the Viterbi Algorithm [10] developed by Andrew J. Viterbi in 1967. First person to try using the Viterbi Algorithm for Maximum Likelihood reconstruction [11] of binary image corrupted by one dimensional blur types and noise was Forney. Burkhardt and Schorb in [9] extended the one-dimensional Viterbi Algorithm to two-dimensional filtering. Later in [12] J. Heanue propose a technique for data detection in a two-dimensional page-access optical memory. Although this method
would provide more optimal results, since the extension of one-dimension to two dimensions had an exponentially growing complexity its implementation was deemed infeasible. To alleviate this problem [13,14] show how to reduce somewhat the computational complexity of 1D and 2D VA respectively. Heanue, Gurkan and Hesselink also described a sub-optimal algorithm, which called Viterbi Algorithm with decision feedback [12]. This non-linear recursive method helps to achieve lower complexity however at times could suffer from error propagation. Another useful contribution to this subject, Chugg [15] derives upper and lower performance bounds for maximum likelihood page detection in the presence of finite-area blur and Gaussian noise. These bounds provide a performance reference by which all maximum likelihood algorithms can be judged [1]. [1] Builds upon the work of [12] and utilizes the bounds of [15] in an effort to improve upon sub-optimal two dimensions Viterbi Algorithm. An alternative contribution which was made by Heanue, Bashaw, and Hesselink is described in [16]. In this research the authors examine the BER performance of various channel codes in a holographic data storage system and discuss the tradeoffs among BER, capacity, and system complexity. In 1996 Neifeld and Chugg presented a novel method [17] for the iterative parallel detection of binary images exposed to two-dimensional blur types.

In this thesis, comparison of the performance of the most mentioned algorithms has been carried out. New methods discussed through the thesis are compared with several known algorithms. The comparisons were based on BER vs SNR graph and bi-level text image restorations. In the comparison between Inverse filter and Wiener filter gray level images were also tested. For ease of understanding, block diagrams are included that display the sequence of steps used in various algorithms. The thesis is organized as follows: Chapter 1 is an introduction to the entire dissertation and includes the literature survey and the problem definitions of the research project. Chapter 2 shows how the degradation model for both continuous and discrete functions can be represented by linear operations. Chapter 3 introduces two types of linear filters used to restore the corrupted image. Description of mathematical approaches for both Inverse and Wiener filters has been done. At the end, appraisals for the performance of both filters are provided. Chapter 4 includes an explanation for both the iterative data detection technique and the first Novel algorithm and explains how the later delivers better
performance than other established algorithms particularly when the used blur is low frequency mask. Chapter 5, the Viterbi algorithm is used as a non-linear recursive approach for image restoration and shows how the image can be modeled by a finite state Markov process. The Viterbi algorithm with decision feedback as a new contribution on the Viterbi algorithm is also discussed in the end of this chapter. In this contribution it is assumed that the data in the rows above the current row were known. Chapter 6 includes a new one-dimensional algorithm based on the mean square error as well as a fixed threshold value. Chapter 7 includes the computer simulation results for all the stated algorithms. Different types of blurs have been used during the simulations to assess the algorithm performances. Finally, Chapter 8 contains the conclusion of this thesis and suggestions for future work.
Chapter 2
DEGRADATION MODEL USED FOR IMAGE RESTORATIONS

2.1 General Degradation Model

Capturing an image exactly as it appears in the real world is very difficult if not impossible. One has to content with Additive White Gaussian Noise (AWGN) and inter-symbol interference (ISI). In case of photography or imaging systems these are caused by the graininess of the emulsion, motion-blur, and camera focus problems. The result of all these degradations is that the image is an approximation of the original. The above mentioned degradation process can adequately be described by a linear spatial model as shown in Fig 2.1. The original input is a two-dimensional (2D) image $f(x,y)$. This image is operated on by the system $H$ and after the addition of $n(x,y)$ one can obtain the degraded image $g(x,y)$. Digital image restoration may be viewed as a process in which we try to obtain an approximation to $f(x,y)$ given $g(x,y)$ and $H$.

![Image degradation model](image.png)

The input-output relationship in Fig 2.1, can be expressed as

$$g(x,y) = H \left[ f(x,y) \right] + n(x,y) \quad (2.1)$$

If the system contains no noise we may assume $n(x,y) = 0$ and $g(x,y) = H [f(x,y)]$. For the linear spatial model depicted in Fig 2.1 we may also write

$$H[k_1 f_1(x,y) + k_2 f_2(x,y)] = k_1 H[f_1(x,y)] + k_2 H[f_2(x,y)] \quad (2.2)$$

and also;

$$H[k_1 f_1(x,y) = k_1 H[f_1(x,y)] \quad (2.3)$$
Equation (2.2) says that the response to the sum of many inputs equals the sum of the response to each individual input. This is known as the additivity property. Equation (2.3) indicates that the response to a constant multiple of any input is equal to the response to that input multiplied by the particular constant. This is referred to as the homogeneity property. The operator $H$ is said to be position or space invariant if

$$H(x-\alpha, y-\beta) = g(x-\alpha, y-\beta)$$

for any $f(x,y)$ and any variables $\alpha$ and $\beta$.

### 2.1.1 Degradation Model for Continuous Functions

Mathematically the degradation model for continuous functions can be expressed as in [3]:

$$f(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x-\alpha, y-\beta) d\alpha d\beta$$

(2.5)

if $n(x,y)=0$ the degraded image becomes

$$g(x,y) = H[f(x,y)] = H\left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \delta(x-\alpha, y-\beta) d\alpha d\beta\right]$$

(2.6)

Since $H$ is a linear operator we can make use of the additivity property and rewrite equation (2.6) as:

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H[f(\alpha, \beta) \delta(x-\alpha, y-\beta)] d\alpha d\beta$$

(2.7)

note that $f(\alpha, \beta)$ is independent of $x$ and $y$ and hence we can write

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) H[\delta(x-\alpha, y-\beta)] d\alpha d\beta$$

(2.8)

The term $h(x, \alpha, y, \beta)$ which is shown below

$$h(x, \alpha, y, \beta) = H[\delta(x-\alpha, y-\beta)]$$

(2.9)

is called the impulse response of the system. That is the response of the system $H$ to an impulse of strength 1 at point $(\alpha, \beta)$. Here the impulse represents a point of light and $h(x, \alpha, y, \beta)$ is known as point spread function (PSF).
If we substitute (2.9) into (2.8) we get
\[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) h(x, \alpha, y, \beta) d\alpha d\beta \] (2.10)

Since \( H \) is position invariant from (2.4) then (2.10) reduces to:
\[ g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) h(x - \alpha, y - \beta) d\alpha d\beta \] (2.11)

Equation (2.11) is a two-dimensional convolution between two matrices \( f(x, y) \) and \( h(x, y) \) and \( H \) can be shorthand noted as:
\[ g(x, y) = f(x, y) \otimes h(x, y) \] (2.12)

where the sign \( \otimes \) denotes the convolution process.

In the presence of Additive White Gaussian Noise the degradation model becomes
\[ g(x, y) = f(x, y) \otimes h(x, y) + n(x, y) \] (2.13)

The degradation that takes place can generally be approximated by linear position invariant processes. However, it is also possible to use non-linear and space variant processes. These are much harder to solve and sometimes may have no known solutions. The Viterbi Algorithm (VA), which will be discussed later in Chapter 5, is an example of such non-linear processing.

2.1.2 Degradation Model for Discrete Functions

A one-dimensional discrete-time model can be easily attained by supposing two functions \( f(x) \) and \( h(x) \) are sampled uniformly to form arrays of dimensions \((1 \times A)\) and \((1 \times B)\) respectively. Here \( x \) will be a discrete variable in the range \([0, 1, \ldots, (A-1)]\) for \( f(x) \) and \([0, 1, \ldots, (B-1)]\) for \( h(x) \). Both \( f(x) \) and \( h(x) \) are periodic and have period \( M \). If \( M \geq (A+B-1) \), the resultant overlap from the convolution can be avoided and extending the functions with zeros to the same length \( M \) one can write as stated in [3] that:
\[ g_c(x) = \sum_{m=0}^{M-1} f_c(m) h_c(x - m) \] (2.14)

If \( x = [0, 1, \ldots, (M-1)] \) then (2.14) can be written in the following matrix form:
\[ g = Hf \] (2.15)

where \( f \) and \( g \) are \((1 \times M)\) column vectors and \( H \) is \((M \times M)\) matrix
For a 2D degradation model the functions \( f(x,y) \) and \( h(x,y) \) are of sizes \((A \times B)\) and \((C \times D)\) respectively. Functions \( f(x,y) \) and \( h(x,y) \) must be padded with zeros to be of size \((M \times N)\) [3].

The convolution of the 2D periodic functions \( f_e(x,y) \) and \( h_e(x,y) \) with periods \( M \) and \( N \) would yield:

\[
g_e(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_e(m,n)h_e(x-m,y-n) + n_e(x,y)
\]  

(2.18)

Here \( n_e(x,y) \) is the AWGN noise with size \((M \times N)\). \( x=[0,1,\ldots,(M-1)] \) and \( y=[0,1,\ldots,(N-1)] \). If we express (2.19) in vector-matrix form we get:

\[
g = Hf + n
\]  

(2.19)

The frequency domain equivalent of (2.19) can be expressed as:

\[
G(u,v) = H(u,v)f(u,v) + N(u,v)
\]  

(2.20)

for \( u = [0,1,\ldots,(M-1)] \) and \( v = [0,1,\ldots,(N-1)] \).
Chapter 3

INVERSE AND WIENER FILTERING TECHNIQUES

3.1 Introduction

The term filter is used to refer to a system that reshapes the frequency components of the input to generate an output signal with some desirable features. Filters (or systems, in general) may be either linear or non-linear. Most basic feature of linear systems is that their behavior is governed by the principle of superposition as discussed in the previous chapter. In particular, a linear system is completely characterized by its impulse response or the Fourier transform of its impulse response, known as the transfer function or PSF. The transfer function of a system at any frequency is equal to its gain at that frequency. Fig 3.1 depicts a general block diagram of a filter emphasizing the purpose for which it is used in different problems. In particular, the filter is used to reshape certain input signals in such a way that its output is a good estimate of the given desired signal. For stationary input and desired signals, minimizing the Mean Square Error (MSE) results in the well-known linear Wiener Filter (WF), which is said to be optimum in the mean-square sense. If one can measure or estimate the transfer function (PSF) of a system, which accurately characterize the system's response, then deconvolution can be readily carried out.

![Schematic Diagram of a Filter](image)

Fig 3.1. Schematic diagram of a filter emphasizing its role in reshaping the input to match a desired signal.

Inverse Filtering is another type of linear filtering technique. Ideally, a Fast Fourier Transform (FFT) would be performed on the image to get $G(F)$ and on the (PSF) to get $H(F)$. By dividing the (FFT) of the degraded image by that of the PSF and taking the
Inverse Fast Fourier Transform (IFFT) of the result, we will get the deconvolved image. This is the approach utilized by an Inverse Filter (IF). Unfortunately, this simple approach is very sensitive to noise, and does not always work in a practical sense. This is due to the division by some very small numbers (equivalent to multiplication by very large numbers). Any noise or uncertainty gets greatly amplified during this process.

### 3.2 Inverse Filter

We have previously seen in Chapter 2 that the noise in the degradation model can be expressed as below:

\[ n = g - Hf \]  

(3.1)

If one does not have any knowledge or assumptions about the noise then, the problem is to seek \( \hat{f} \) such that \( H\hat{f} \) approximates \( g \) in a least square criterion. In other words one need to minimize the square errors in the manner described by [3]:

\[ \| n \|^2 = \| g - H\hat{f} \|^2 \]  

(3.2)

The length of the noise vector is \((1 \times L)\) and for real numbers \( \| n \|^2 = n^T n \). Where \( T \) indicates the transpose of a matrix. Note also \( \| g = H\hat{f} \|^2 = (g - H\hat{f})^T (g - H\hat{f}) \).

If we use \( W(\hat{f}) \) to denote the norm of \( (g - \hat{h}f) \) then:

\[ W(\hat{f}) = \| g - H\hat{f} \|^2 \]  

(3.3)

Equation (3.3) implies that we can minimize LSE as a function of the estimated image \( \hat{f} \). Minimizing (3.3) can be done by differentiating \( W \) with respect to \( \hat{f} \) and by setting the result equal to the zero vector \( \theta \):

\[ \frac{dW(\hat{f})}{d\hat{f}} = \| g - H\hat{f} \|^2 = -2H^T (g - H\hat{f}) = \theta \]  

(3.4)

Solving for \( \hat{f} \) in (3.4) yields:

\[ \hat{f} = (H^H)^{-1} H^T g \]  

(3.5)

If \( H^{-1} \) exist and \( H \) is a square matrix \((N = M)\) (3.5) reduces to:

\[ \hat{f} = H^{-1} (H^T)^{-1} H^T g = H^{-1} g \]  

(3.6)
The equivalent frequency domain version of (3.6) becomes:

\[ \hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \quad u,v=0,1, \ldots, (N-1) \]  

(3.7)

If we consider \( H(u,v) \) as a filter response that multiplies \( F(u,v) \) to produce the degraded image \( G(u,v) \) then by taking the IFFT for \( \hat{F}(u,v) \) we can get the approximation of the original image \( g(x,y) \) where \( y,x=[0,1,2, \ldots, (N-1)] \).

\[ \hat{f}(x,y) = \text{IFFT} \left[ \hat{F}(U,V) \right] = \text{IFFT} \left[ \frac{G(u,v)}{H(u,v)} \right] \]  

(3.8)

Note that if \( H(u,v) \) has small values for any “u” plane then, the restoration process would become difficult. To avoid this one can generally neglect these small values from the \( H(u,v) \) matrix without really affecting the restoration process. If we substitute (3.7) into (2.20) we will get:

\[ \hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \]  

(3.9)

It is clear for (3.9) that if the values of \( H(u,v) \) are zeros or very small, the original image \( F(u,v) \) will be degraded more. In practice, the values of \( H(u,v) \) often drops off rapidly as a function of distance from the origin. So in order to avoid these small values the restorations are generally carried out in a region around the origin. If we assume that the image \( f(x,y) \) is a unit impulse function, then the degraded image equals the transfer function of the system because the Fourier transform of a unit impulse function equals unity [3]:

\[ G(u,v) = H(u,v)F(u,v) \approx H(u,v) \]  

(3.10)

\[ \text{FFT}[\delta(x,y)]=1 \]

The most important point to note here is that by equating the original image to a unit impulse function one can obtain an approximation to the transfer function \( H(u,v) \) of the system. Fig 3.2-(a) below shows a point image \( f(x,y) \) and Fig 3.2-(b) shows the degraded output image, which approximately equals the transfer function \( H(u,v) \) of the system.
By using (3.8) for close values of $u$ and $v$ to the origin of the $uv$ plane, we will get the restored image shown in Fig 3.3-(c). If one increases the region of the values $u$ and $v$, the restored image will be as shown in figure Fig 3.3-(d). This version of the restored image is worst due to the difficulties introduced by small and vanishing values of $H(u,v)$ as discussed previously.

Fig 3.3. Image restoration by using Inverse filter at different values of $H(u,v)$. 
If \( H(u,v) \), \( G(u,v) \), and \( N(u,v) \) are known, then an exact inverse filtering expression can be obtained from (2.20) as shown below:

\[
F(u,v) = \frac{G(u,v)}{H(u,v)} - \frac{N(u,v)}{H(u,v)}
\]  

(3.11)

### 3.3 Wiener Filter

Wiener filter's working principle is based on the least squares restoration problem. The technique aims to minimize the functions of the form \( \| \beta \, \hat{f} \|^2 \), where \( \beta \) is a variable number. For different values of \( \beta \) we can introduce different solutions to the constraint equation \( \| g - H\hat{f} \|^2 = \| n \|^2 \). By using Lagrange multiplier approach, we can express the last equation in the following form:

\[
J(\hat{f}) = \| \beta \, \hat{f} \|^2 + \alpha(\| g - H\hat{f} \|^2 - \| n \|^2)
\]

(3.12)

Where \( \alpha \) is a constant called the Lagrange multiplier. Differentiating (3.12) with respect to \( \hat{f} \) and equating the result to the zero vector \( \theta \) one can obtain the desired solution as a function of \( \hat{f} \):

\[
f = (H^T H + \gamma \beta^T \beta)^{-1} H^T g
\]

(3.13)

where \( \gamma = 1/\alpha \). Let

\[
\beta^T \beta = R_f^{-1} R_n , \quad R_f = E(ff^T) , \quad R_n = E(nn^T).
\]

(3.14)

The operator \( E \) denotes the expected value operation and \( n, f \) are as previously defined in section 2.2.2. Substituting (3.14) into (3.13) and carrying out some simplifications discussed in detail in [3] one can obtain Wiener filter's mathematical representation as shown below:

\[
\hat{F}(u,v) = \left[ \frac{H^*(u,v)}{H(u,v)^2 + \gamma [S_n(u,v)/S_f(u,v)]} \right] G(u,v)
\]

(3.15)

Here \( u,v = [0,1,2,3…(N-1)] \). \( S_n(u,v) \) and \( S_f(u,v) \) are called the power spectrum (spectral density) of \( f(x,y) \) and \( n(x,y) \) and \( H \) is a square matrix. When \( \gamma = 1 \) (3.15) is known as Wiener filter otherwise, when \( \gamma \) is variable this expression is called the Parametric Wiener filter. Note that when \( \gamma = 1 \) we obtain the optimal resorted image for which the
quantity \( \{ E[F(u,v) - \hat{F}(u,v)]^2 \} \) is minimum. When the ratio \( S_n(u,v)/S_j(u,v) \) is unknown it considered to be a constant value, \( K \), and (3.15) reduces to:

\[
\hat{F}(u,v) = \left[ \frac{H^*(u,v)}{H(u,v)^2 + K} \right] G(u,v) \quad K = S_n(u,v)/S_j(u,v)
\] (3.16)

In (3.16) \( K \) it may assume values such as 0.01, 0.1, 0.05. Sometimes it is taken as \( 2 \sigma^2 \) where \( \sigma^2 \) is the variance of the noise. Simulation results indicate that as the \( K \) value decreases the amount of noise in the restored images grew and at \( K = 0 \) this value reached its maximum. Here it is important to notice that Wiener filter with \( K = 0 \) is equivalent to the inverse filter. Also note that \( K \) is directly dependent on the signal to noise ratio selected.
Chapter 4

ITERATIVE DATA DETECTION TECHNIQUE

4.1 Introduction

M. A. Neifeld, K. M. Chugg, and B. A. King first introduced the iterative data detection technique in 1996. This novel two-dimensional technique was intended for reliable detection in page oriented optical memories (POOM). It was showed that the algorithm could offer significant performance improvements over other well-established techniques like Wiener filtering and simple threshold detection.

The novel 2D method explained herein is motivated by the decision feedback technique, first introduced by Alexander Dual Hellen, and it can also be implemented in hardware in a parallel fashion. The first part of this chapter will explain details of the 2D de-blurring technique as proposed by Neifeld and Chugg and in the second part a variation which also turns out to be a novel-method delivering better performance will be discussed.

4.2 Page Oriented Optical Memory

A great deal of effort has been put in while investigating the page access optical memories in order to increase storage capacity, decrease the access time and facilitate high aggregate data rates, where the input and output data are transferred in parallel on page by page basis. POOM is a technique for storing multiple bits of information at a single location in a crystal. This technique exploits the fact that molecules in crystals absorb and radiate light at many different frequencies, and has the potential of storing 1-million bits in a one-cubic-micron spots in a crystal. To write information, a series of laser pulses (constituting a binary message) burns a spectral hole at a tiny spot in a crystal. To read the information, a second series of pulses causes the crystal molecules to radiate a frequency pattern identical to that of the first pulse. Recently, 1600 bits was written to a single 100-micron spot in a crystal. With a few advances in technology, it may be possible to store 50 million bits in the spot and read the information at a blazingly fast rate like 40 million bits per second. Furthermore, any deviation from ideal imaging
during writing or reading processes results in inter-pixel cross talk and a higher output error. The term inter symbol interference (ISI) is used when the pixel data value is affected by the neighborhood data bits. In the case of ideal imaging, the system is free of ISI and the detection may be performed using threshold detection for each bit separately. The subsection below focus on the data detection and present the iterative detection algorithm used to improve the detection performance in the presence of 2D intersymbol interference (ISI) and noise.

4.2.1 Algorithm Description

ISI of optical channels are similar to point spread function (PSF) of imaging systems. A point spread function may be characterized in the time domain by a matrix $H$ of size $(2K+1)\times(2K+1)$. For bi-level images one may assume an intensity array $P$ where the values $p_{ij}$ are either 0 or 1. The received intensity array elements $r(i,j)$ are the result of a convolution between the PSF matrix $h(m,n)$ and the input elements $P(i,j)$:

$$r(i,j) = \sum_{m=-K}^{K} \sum_{n=-K}^{K} h(m,n) p(i-m, j-n)$$

where $i, j = [1,2,3\ldots N]$, and the image size is $(N\times N)$. The PSF of a radially symmetric Gaussian blur spot with standard deviation $\sigma_b$ may be computed using the expression shown below:

$$h(i, j) = \int_{-\sigma_b}^{\sigma_b} \int_{-\sigma_b}^{\sigma_b} e^{-\frac{x^2+y^2}{2\sigma_b^2}} \, dx \, dy$$

If $\sigma_b=0.5$ and $K=1$ Then the corresponding values of PSF are $h(0,0) = 0.466$, $h(-1,-1)=h(1,1)=0.025$ and $h(0,1) = h(1,0) = h(0,-1) = h(-1,0) = 0.107$. Note that the summation of PSF elements equals unity. After the addition of Gaussian noise the received array becomes:

$$A = \alpha (i,j) = r(i,j) + n(i,j)$$
here, \( n(i,j) \) is AWGN with zero mean and variance \( \sigma_n^2 \). The signal to noise ratio (SNR) of the received electrical signal can be defined as

\[
\text{SNR} = \frac{2E[r^2(i,j)]}{E[n^2(i,j)]} = \sum_{j} \frac{h^2(i,j)}{\sigma_n^2}
\]

(4.4)

Where \( E\{\cdot\} \) denotes the expectation operator and the assumption of identical, independent distribution for all the data has been made. For data-detection it is possible to define two performance bounds. The first one is based on a fixed threshold detection approach, which delivers the worst case bound. With this method the threshold \( TH \) can be calculated by taking the average effect of received samples that were exposed to ISI.

\[
TH = \frac{[E[r(i,j)|p(i,j) = 1] + E[r(i,j)|p(i,j) = 0]]}{2}
\]

(4.5)

The second bound, which is the maximum-likelihood detection solution, represents all possible received pages in the presence of ISI. If the received page has \((N \times N)\) size the number of possible states will be \( 2^{N^2} \). After the convolution between each state and PSF the one which has the least square error is considered to be the optimum page. Though for simulations searching through \( 2^{N^2} \) possible values are okay such a procedure in practice is not feasible.

The steps of the iterative detection algorithm proposed in [17] may be summarized as shown by Fig 4.1:

1- Apply the fixed threshold \( Th \) mentioned above to the output from Wiener filter that will produce data \( P(i,j) \). If the output from Wiener filter is more than the \( Th \) value the pixel \( p_{ij} \) will be zero, otherwise it will be a one.

2- Assuming that each pixel \( p_{ij} \) may become either a zero or a one and for both cases produce two new matrices, \( Y \) and \( YY \). Here \( Y = P \) and \( YY = (1-P) \) as shown in Fig 4.1.

3- Convolve each pixel in the \( Y \) and \( YY \) matrices (together with their eight neighborhood pixels) with the optical system response PSF, to obtain two new matrices \( r_Y(i,j) \) and \( r_{YY}(i,j) \).

4- Then the differences \(|r_Y(i,j) - \alpha(i,j)|\) and \(|r_{YY}(i,j) - \alpha(i,j)|\) are both computed and if \(|r_Y(i,j) - \alpha(i,j)| < |r_{YY}(i,j) - \alpha(i,j)|\) then the optimum restored data
which best matches to the original is \( r_y(i, j) \) otherwise it will be \( r_{yy}(i, j) \). For \( r_y(i, j) \) and \( r_{yy}(i, j) \) the optimum estimated pixel will be \( Y_{i,j}, YY_{i,j} \) respectively.

5- Repeat the same procedure for all pixel values in the data page. Updating of \( r_y(i, j) \) and \( r_{yy}(i, j) \) take place simultaneously at each iteration.

The method described in [17] points out that since 86% of the time, simulations have converged after two iterations (for a page size of 128 \( \times \) 128) then it is possible to develop an efficient hardware implementation of the algorithm. The following diagram is a block wise representation for the iterative data detection technique.

![Block diagram representation for Iterative Data Detection.](image)

**4.3 Double Threshold Based Novel De-blurring Technique**

The novel method [18], discussed here is an algorithm used for improving the performance of data-detection proposed by Neifeld and Chugg [17]. The proposed new algorithm which is a variation of the iterative data-detection method is also linear filter based however it uses two Wiener filters and two separate threshold values, one for each Wiener filter.
4.3.1 New Algorithm Description

Steps of the novel algorithm proposed can be listed as follows:

1- Get an output image $Y$, from the Wiener filter.

2- Threshold the output data $Y$ by using a fixed threshold $\text{Th} = 0.5$. Call it the new matrix $YY$.

3- Convolve the thresholded data $YY$ with either a copy of the entire or a copy of the truncated version of the impulse response of the former Wiener filter, $h_w$. Call the result $ZZ$.

4- Apply to $ZZ$ matrix the second threshold which in this study was taken as 0.042 and call the result $KK$.

5- Finally use Neifeld's decision criteria by creating two new matrices, convolving both with the PSF and then comparing the absolute value of the differences of the convolution results and the original data. As a result, another matrix represents an enhanced version of the original image will be obtained see Fig 4.2.

In this version of the algorithm the second Wiener filter is used to enhance the output of the former one by boosting up various frequency components. The second threshold is used for separating the low frequency components (represent the black pixels in the image) from the rest. If in steps-5 and 2 of the algorithm the chosen PSF is a low frequency mask (PSF) (i.e. one with elements that are all positive valued) then in the resulting image the white pixels will be dominant and black pixels will be far less in number. In such an image black color represents the original information bearing data while the white color represents the background of the image. On contrast if in steps-5 and 2 a high frequency mask is used (i.e. sum of all pixel values equals to zero and has negative elements) the black color will be dominant and white color will be representing the original data. The majority of black color in the output image when a high frequency mask is used due to the output frequency components from the former Wiener filter is very small, hence most of them will change to zeros after the first and second thresholds. But this algorithm will offer better performance than others do (in the case of high frequency mask) if the used thresholds are adjusted properly. Chapter-7 will show that this method is an enhanced version of the iterative data detection and delivers better performance than the previously discussed techniques.
Fig 4.2. Double threshold-based architecture.
Chapter 5

BILEVEL IMAGE RESTORATION VIA

2D VITERBI ALGORITHM

5.1 Introduction

In many restoration problems, the \textit{a-priori} knowledge of pixel amplitudes of the original image is defined to be bi-level. This chapter shows how to incorporate this knowledge into optimal image reconstruction. The application of linear filter based methods such as Wiener or Kalman filtering are non-optimal for the task of image reconstruction because the \textit{a-priori} knowledge of the binary valued original image cannot be incorporated into the solution. Such techniques also show a clear trade-off between signal improvement and noise enhancement. The Viterbi Algorithm (VA) utilizes the principle of dynamic programming to achieve Maximum-A-posteriori (MAP) probability data detection in dispersive digital communication channels with known transfer characteristics. The resulting nonlinear recursive filter provides the optimal solution with a superior performance at the expense of increased detector complexity. It is possible to extend the VA to a two-dimensional image restoration on the basis of a maximum-a-posteriori criterion for the problems in the form of a finite discrete set of amplitudes of the original image. The image distortions are modeled with a two-dimensional finite state Markov model in analogy to a communication channel. In the eighties, H. Burkhardt proposed a nonlinear technique known as VA with Decision Feedback (VA-DF),[12]. VA-DF technique not only would provide certain performance improvements it would also achieve this at a low complexity. The improvements were achieved by compensating for intersymbol interference using tentative decision feedback. Though generally it would seem to work okay at times wrong decisions could be fed back which then could cause the system to suffer from error propagation.
The objective of this chapter is to apply the VA to the received data, which has been corrupted by spatial intersymbol interference and additive white Gaussian noise. It was assumed that the intersymbol interference would affect only one dimension as in a low track density. For high track densities as in optical storage media a crosstalk is produced between the tracks and this dispersion is known as two-dimensional intersymbol interference.

5.2 Problem statement

A continuous digital channel with a spatially limited PSF can be modeled by a finite state discrete-time Markov process observed in memoryless noise. The Viterbi algorithm is a recursive optimal solution to the problem of maximum a posteriori probability estimation of the state sequence of a discrete-time finite-state Markov process observed in memoryless noise. A discrete time Markov process may be characterized as follows. The state $x_k$ at time $k$ is one of a finite number of states $M$ (e.g. the state space $X$ has $\{1, 2, 3, \ldots, M\}$ states). Assume at time $t_0$ the initial state is $x_0$ and at time $k$ there is the final state $x_k$, where these two states are known. So we can represent the state sequence by a finite vector $x = \{x_0, x_1, \ldots, x_k\}$. The probability to be in state $x_{k+1}$ at time $k+1$, given all states up to time $k$, depends only on the state $x_k$ at time $k$:

$$P(x_{k+1} | x_0, x_1, \ldots, x_k ) = p(x_{k+1} | x_k ) \quad (5.1)$$

The transition probabilities $\xi_k = p(x_{k+1} | x_k )$ is the probability of being in state $x_k$ and at time $k+1$ one will shift to state $x_{k+1}$. It is convenient to define the transition $\xi_k$ at time $k$ as the pair of states as follow:

$$\xi(k) = (x_{k+1}, x_k ) \quad (5.2)$$

and the transition probability can be defined by:

$$p(x_{k+1} = j | x_k = i ) = p_{ij} \quad (5.3)$$

where $\sum_j p_{ij} = 1$. From (5.3) one can specify one-step transition probability matrix as:
\[ P = \begin{bmatrix}
P_{00} & P_{01} & P_{02} & \cdots & P_{0j} \\
P_{10} & P_{11} & P_{12} & \cdots & P_{1j} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
P_{k0} & P_{k1} & P_{k2} & \cdots & P_{kj}
\end{bmatrix} \tag{5.4}
\]

The process is assumed to be observed in memoryless noise. That is, there is sequence \( z \) of observations \( z_k \) in which \( z_k \) depends probabilistically only on the transition \( \xi_k \) at time \( k \) and \( \xi = (\xi_0, \xi_1, \ldots, \xi_{K-1}) \)

\[
p(z \mid x) = p(z \mid \xi) = \prod_{k=1}^{K} p(z_k \mid \xi_k) \tag{5.5}
\]

In fact we have two cases, the first case when the observed sequence \( z_k \) depends only on the state \( x_k \) of the transition \( \xi_k \) and the second case in which \( z_k \) depends probabilistically on an output \( y_k \) at time \( k \), where \( y_k \) is the output from a deterministic function of the transition \( \xi_k \).

\[
p(z \mid x) = \prod_{k=1}^{K} p(z_k \mid x_k) \tag{5.6}
\]

Deterministic function can be an XOR gate, usually used with convolutional coding. The present and \( v \) previous inputs to the deterministic function determine the observed output \( z_k \) [11].

\[
y_k = f(u_k, \ldots, u_{k-v}) \tag{5.7}
\]

Where \( u \) is the input sequence shown below:

\[
u = (u_0, u_1, \ldots) \tag{5.8}
\]

The observed sequence \( z \) is the output of a memoryless channel whose input is \( y \). This process can be modeled by a shift register of length \( v \) with inputs \( u_k \). Finally, we state the problem to which the VA is a solution:

Given a sequence \( z \) of observations of a discrete-time finite-state Markov process in memoryless noise, find the state sequence \( x \) for which the a posteriori probability \( p(x \mid z) \) is maximum. Alternately, find the transition sequence \( \xi \) for which \( p(\xi \mid z) \) is maximum.
5.3 Finding the best path in the trellis

The degradation that takes place in imaging system is similar to degradation that would occur on a data sequence transmitted through an AWGN channel. At each time $k$, the $k^{th}$ data symbol $d_k$ is transmitted. And at the receiver the sample $r_k$ is received. If the system transmits binary data in-groups of $n$ bits, then the symbol alphabet consists of $N=2^n$ distinct symbols. The operation of an ISI-corrupted system can be visualized by means of a trellis diagram like the one shown in Fig 5.1. At each time $k$ the system is in one of $W$ possible states, $S_i$, where $i=[1,2,...,W]$.

![Trellis Diagram](image)

If the ISI has memory of length $L$, the current state of the system will depend on the value of the last $L$ symbols, $d_{k-1},...,d_{k-L}$. The value of $L$ refers to the constraint length or the number of data symbols that affect a given output. If ISI has memory $L$, then any output will be affected by $(L+1)$ symbols. During $k^{th}$ time interval, the system transmits a new data symbol and makes a transition to another state in the next stage of the trellis. In the absence of noise one can detect the original signal $S_i$ at any given time using the initial state, the destination state and the known ISI characteristics. As data symbols are transmitted, the system traces the paths through the trellis. The problem consists of finding the most likely path, given the most correct data $d$. If the sequence of received
data is \( r = (r_1, \ldots, r_k) \) then, the optimum sequence detector chooses the data sequence \( d = (d_1, \ldots, d_k) \) that maximizes the conditional probability expressed below:

\[
p(r \mid d) = \prod_{k=1}^{K} p(r_k \mid d_{k-L}, \ldots, d_k) = \prod_{k=1}^{K} p(z_k \mid x_k)
\]

where, \((L+1)\) data values \( d_{k-L}, \ldots, d_k \) determine the state \( x \) in the trellis at time \( k \). Thus the detection problem can be expressed as finding the path through the trellis that maximizes the product:

\[
\prod_{k=1}^{K} p(r_k \mid S_{k,p})
\]

where \( S_{k,p} \) is the output signal of the transition at time \( k \) of \( p^{th} \) possible path. As the channel is AWGN, it is possible to rewrite the probability density function in (5.10) in the following way:

\[
p(r_k \mid S_{k,p}) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left[ -\frac{(r_k - S_{k,p})^2}{2\sigma^2} \right]
\]

where, \( \sigma^2 \) is the variance of the AWGN. If (5.11) is inserted into (5.10) and the logarithm of the product in (5.10) is taken the path that would be correct would be the one minimizing the summation below:

\[
D_p(r) = \sum_{k=1}^{K} (r_k - S_{k,p})^2
\]

where \( D_p \) is the Euclidean distance between the received sequence and one of the \( p^{th} \) path. The value \( D_p \) is the measure, which the Viterbi detector bases its decisions on.

Suppose one wants to determine the, *minimum distance path*, which passes through a given state \( S_{j+1} \) at time \( t_k \). There are \( N \) paths entering the state \( S_{j+1} \) at time \( t \) from state \( S_j \). Each of these paths has a distance metric \( D_{N} \). The total paths for one stage or time are \( D_{pa} \) measured by (5.9). The subscript 'a' can take on values 1, \ldots, \( W \) where \( W \) denotes the number of states. A path has the smallest metric is referred to as a survivor path. There are \( K \) survivor paths in all the trellis, one for each received data. From all possible paths in the trellis there is only one survivor path through the entire trellis that has the
minimum accumulative distance $\sum_{k=1}^{K} D_k$, where $D_k$ is the Euclidean distance of the survivor path in each stage.

5.4 Extension of the 1D-VA to 2D problems

Model for the two-dimensional (2D) digital channel is shown in Fig 5.2. 2D array of discrete values $f(i, j)$ is transmitted through the channel. The values $h(i, j)$ represents the channel impulse response or in frequency domain the transfer function of the system. The transfer function is assumed to be finite and represented by an $(m \times n)$ matrix. The sequence values $n(i, j)$ represent the Additive White Gaussian Noise. If the system is linear and time invariant as mentioned in Chapter 2, then the output sequence $g(i, j)$ is equal to $f(i, j) \otimes h(i, j) + n(i, j)$, where the sign $\otimes$ denotes the convolution process. This degraded version of the image $g(i, j)$ then constitutes the input to a 2D Viterbi detector to produce an estimate $\hat{f}(i, j)$ of the original data sequence.

![Fig 5.2. 2D degradation model.](image)

Burkhardt explained in [12] how one could extend the 1D-VA to 2D-problems and this is briefly stated below. Consider a system with transfer function represented by $3 \times 3$ PSF. Selecting the location of the noise free pixel will depend not only on the tested pixel but as well as on the surrounding neighbours of the tested pixel. One can define the symbol alphabet as the number of distinct values that can be taken from a column whose elements equal to the elements of the transfer function. Using a $(3 \times 3)$ transfer function, one can obtain eight distinct states using the three bits of the last column. The width of
transfer function determines the length of ISI memory. For the 3×3 transfer function, the length of the memory that affect the output value equals \( L+1=3 \). This implies that ISI has memory of length \( L=2 \). Figure 5.3 depicts a portion of a two dimensional page or data. The solid box encloses the bits that define the initial state and the dashed box encloses the bits that define the destination (final) state. The black dots correspond to the detected bit due to the transition from the initial to the final state. If the initial state is denoted by \( x^k \) and the final state as \( x^{k+1} \) then we can say that the 2D VA technique depends on a finite-state Markov process. That is, the probability of being in state \( x^{k+1} \) at time \( (k+1) \) is dependent only on the immediate preceding state \( x^k \) at time \( k \) and not on the other previous states \( \{x^0, x^1, \ldots, x^{k-1}\} \).

\[
p(x^{k+1} | x^k, \ldots, x^0) = p(x^{k+1} | x^k)
\]  

(5.13)

The noise free received output bit depends on the Euclidean distance metric. The Euclidean distance metric is calculated by the comparison of the received value \( r(i, j) \) and the expected new value \( y(i,j) = f(i,j) \otimes h \). Here \( f(i,j) \) is a state composed of the present state and the second column of one of all-possible transition (final) states. The state \( f(i,j) \) is considered as the estimated state \( \hat{f}(i,j) \) if it gives the least Euclidean distance metric. In other word, the estimated state \( \hat{f}(i,j) \) must give

\[
\text{Min} \{y(i,j) - r(i,j)\}^2
\]

(5.14)

For the portion of the image depicted in Fig 5.3, the total number of possible states for \( f(i,j) \) is 64 states. In general, for each 3-bit output symbol, the central bit is chosen as the noise free detected bit. Detection continues on a row-by-row basis until the entire data page has been estimated. This technique can be extended in a straightforward manner to systems with larger transfer functions by increase the number of states and the size of the symbol alphabet. One main problem faced by this detection technique is that error propagation may lead to unacceptable performance. Yet another disadvantage is that in the above technique decisions are based on the observation of a progression of symbols in only one dimension. Information about the symbol sequence in the vertical dimension is
not used. Finally, we can say that the possible number of states per transition may results in a high-complexity Viterbi detector as the ISI length grows. The computational complexity of the VA is proportional to the number of states, which was stated above to be equal $M^L$.

![Diagram of state definition](image)

Fig 5.3. State definition for a system with $(3\times3)$ transfer function using the VA detection.

### 5.5 Viterbi Detection with Decision Feedback

A lower complexity detector than the 2D-VA is possible if one allows feedback of the estimated data. The idea of Decision Feedback (DF) was first stated in [19] and later used by [12] to reduce the complexity of the 2D-VA. It was assumed that the upper row was known and the next row was estimated under the assumption that data in the first row was detected correctly see Fig 5.4. The effect of the first row was then subtracted from the received data and the Viterbi detection would follow for other rows. For example for a $(3\times3)$ transfer function in a 2D-VA the total number of states is 64 however for the 2D-VA with DF this number reduces to 16. This implies a reduction by a factor of four in the detector complexity. The reduced complexity is attained at the expense of the memory required for storing rows of previously detected data. Another disadvantage of DF is that error propagation can result; however, in most cases the propagation effects are not enough to eliminate the BER gains achieved when feedback is incorporated.
Fig 5.4. State definition for a system with (3×3) transfer function using the VA-DF principle.
Chapter 6

NOVEL ONE OR TWO DIMENSIONAL DE-BLURRING ALGORITHMS BASED ON MINIMUM MEAN SQUARED ERROR AND A SELECTED FIXED THRESHOLD

In this study we investigate the degree to which a new de-blurring algorithm based on the Minimum Mean Squared Error (MMSE) and a fixed threshold “Th”, can successfully restore images that have been degraded by Additive White Gaussian Noise (AWGN) and various blur types. The new method proposed herein is used for de-blurring images exposed to 1D point spread functions (PSF) [20]. However, the chapter also points out how the 1D approach can be extended to two dimensions. The simulation results are compared with those of other well-established methods in the literature (presented in chapter-7). Attained results indicate that the suggested technique either performs better than some known methods or compares well to them. It is observed that for moderate to high signal to noise ratios the proposed algorithm would provide substantial improvements in BER performance when compared to hard or soft decision VA-techniques for AWGN and Raleigh Fading channels.

6.1 Markov Process Modeling

A continuous digital channel with a limited extend point spread function can be modeled by a discrete channel model with AWGN. One can describe this discrete channel by a finite-state discrete-time Markov process observed in memoryless noise. It is assumed that the original data has \( b \) possible discrete amplitude levels (e.g., \( b = 2 \) for binary data). The data degradation is caused by a PSF of dimension \((m \times n)\). Measured data in an observation space \( Y \in y \) of dimension \((M \times N)\) can be defined by signals in an area of dimension \((M+m-1) \times (N+n-1)\). Therefore one can get:
\[ X \in \mathbb{R}^{(M+m-1)\times(N+n-1)} \]
\[ H \in \mathbb{R}^{m\times n} \]
\[ Y, Z \in \mathbb{R}^{M\times N} \]

where, \( X \) is the original data, \( H \) is the finite dispersion function and \( Y \) is the measured data. The states of the Markov chain are given by stripes of data of dimensions \((M+m-1)\times(n-1)\) and its transitions \(w_k^k\) pairs) are defined by consecutive pairs of states, see Fig.6.1 and Fig.6.2.

\[ w_k = (x^{k-1}, x^k) \]
\[ \dim w_k = (M + m - 1) \times n \]
\[ \dim x^k = (M + m - 1) \times (n - 1) \]

The transitions and the Markov states can be related as shown below:

\[ X = [w^1, w^2, ..., w^N] = [x^0, x^1, x^2, ..., x^N] \]  \hspace{1cm} (6.3)

![Fig 6.1. The transmitted data in the original space X and the observation space Y,Z for a 3x3 PSF.](image)

A Markov process requires that the probability to be in state \( x^{k+1} \) depends only on the present state and is independent of all other previous states \( \{x^0, x^1, ..., x^{k-1}\} \), that is:

\[ P(x^{k+1} \mid x^k, ..., x^1, x^0) = P(x^{k+1} \mid x^k) \]  \hspace{1cm} (6.4)
Fig 6.2. Forming states and transitions from two dimensional data.

There are overlapping segments between two consecutive pairs of Markov states for the 1D version and overlapping areas in the two-dimensional spatial domain. The observation sequence $Z = [z^1, z^2, ..., z^N] \in \mathbb{R}^{M \times N}$ depends probabilistically only on the transition sequence $w^k$:

$$P(Z | X) = P(Z | W) = \prod_{k=1}^{N} P(z^k | w^k) \quad (6.5)$$

Hence the problem of optimal data detection at different SNR values can be stated as:

*If we receive a sequence of sampled columns of the degraded data, find the state sequence $X = x^k$, which maximizes the conditional probability $P(X | Z)$.*

This problem is equivalent to finding the shortest path through a decision trellis with weights proportional to the following expression [2]:

$$\lambda(w^k) = -\ln P(x^{k+1}, x^k) - \ln P(z^k | w^k) \quad (6.6)$$

### 6.2 One Dimensional Novel De-Blurring Algorithm

The first step of the newly suggested de-blurring algorithm is to convert any given bi-level image into an information frame as depicted in Fig 6.3-(a) and then to divide this frame into equal length stripes. For one-dimensional frames, each stripe may be divided into two or more bits long. In our study we have taken this length as three bits long. Since it is possible to artificially blur an original image by performing convolution between the image and a selected point spreads function, all three bit stripes would be convolved with
the row-vector as shown in Fig 6.4 and to these values AWGN is added and noised samples are denoted by \( Y \).

![Diagram of row-vector and AWGN addition](image)

Fig 6.3. Partitioning 1D and 2D data into smaller stripes or areas.

While finding the minimum mean squared error one first needs to think of all the possible three-bit combinations that could be the actual bits in the first stripe and then to convolve each one of these with the \((1 \times 3)\) PSF. For three bit stripes we have eight different combinations, \(2^3\).

![Diagram of mean squared error computations](image)

Fig 6.4. Mean squared error computations and thresholding.
If we denote the result of the artificial convolution as $X$, then the minimum mean squared error values are attained by computing $[Y - X]^2$. Then the next step is to choose a proper threshold $Th$. Usually a small threshold delivers efficient performance particularly at high signal to noise ratios. From the possible three bit state combinations used in computing the $[Y - X]^2$, values which lead to a mean squared error value below the selected fixed threshold are then used to form a trellis. With this method it is possible that only few of the three bit combinations would satisfy the threshold criteria. If the numbers of possible three-bit combinations (states) that satisfy the threshold criteria for the first received stripe (first stage in trellis) is $z_1$, and this number increases to $z_2$ for the second received stripe (second stage) then the difference of states, $(z_2 - z_1)$, are added as zero states (000) to the stages which have lower number of states. This process is repeated for all the stages in the trellis until the last stripe is received. Figure 6.5, below shows the process being applied to the first four stages of the trellis:

<table>
<thead>
<tr>
<th>1st stage</th>
<th>2nd stage</th>
<th>3rd stage</th>
<th>4th stage</th>
<th>5th stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>111</td>
<td>000</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>001</td>
<td>110</td>
<td>111</td>
<td>..........</td>
</tr>
<tr>
<td>101</td>
<td>010</td>
<td>000</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>000</td>
<td>101</td>
<td>000</td>
<td>010</td>
<td>..........</td>
</tr>
<tr>
<td>000</td>
<td>000</td>
<td>000</td>
<td>110</td>
<td>..........</td>
</tr>
</tbody>
</table>

Fig 6.5. Matching number of states in different stages of the trellis.

Note that the threshold is chosen small in order to select the three-bit combinations that would give small mean squared error values. Since in most cases we would have limited number of states satisfying the threshold criteria then the tracing process through the trellis will be over these states only and not include all the eight combinations as in Viterbi Algorithm (VA). The depth of the trellis is directly related with the length of the information frame. While tracing through the trellis the aim is to select the path with the least accumulated error. Another important point is tracing the same overlapping bits.
between two consequent three-bit stripes, when searching through the trellis. The
searching process that is carried out can be described as follows:
Let us assume that each bit in the frame of data has a location pointer and starting from
the second left value we denote each value as \(x_1, x_2, x_3, x_4, \ldots\) as in Fig. 6.6 below:

\[
\begin{array}{ccccccccccc}
X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\end{array}
\]

As seen in the above figure the second and third bits in the first stripe \((x_1, x_2)\) have an
overlap with the first and second bits of the second stripe \((x_3, x_4)\). The key to searching the
trellis is to check if these overlapping bits are equal. Another important point is to travel
through the trellis to the further possible depth while following these overlapping bits.
The process follow the searching criteria through the stages of the trellis by adding the
length of the stripe (in our case \(n=3\)) to the initial values \([x_1, x_2]\) in the first stage and to
\([x_3, x_4]\) in the second stage. As a result you will obtain the next four overlapping values
\([x_4, x_5]\) and \([x_6, x_7]\). In general \(x_i = x_{i+3}\) for \(i = 1, 2, 3, 4, \ldots\). When the entire processing is
finished for each stage of the optimum path the central bit is chosen as an estimate for the
original transmitted bit. The number of stages equals the number of stripes and also
equals \((r/n)\), where \(r\) is the size of the received data and \(n\) is the length of each stripe. We
have assumed in this work that the first three bits of the information frame is a preamble
inserted by us. This is to initiate the algorithm and for increasing its efficiency. As
depicted in Fig 6.3-(b) it is also possible to extend this new idea to two dimensions. In the
2D version, an image of size \((M\times N)\) is first partitioned into smaller blocks of size \((m\times n)\). If we call the overlapping region between two consecutive blocks \(L\), the region \(L\) would
be of size \((m \times z)\). Also with the extension to 2D the possible number of states would rise from four (in 1D) to \(2^{mn}\). The remaining steps of the algorithm are same as its 1D version. Let us assume that the data to be transmitted is :

\[
\begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}
\]  

(6.7)

With the addition of the preamble \([1 1]\), the data frame will be as follows:

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}
\]  

(6.8)

Then assume that after the degradation by ISI and AWGN (\(\sigma = 0.0839\)) the received sequence is

\[
Y = [1.0412\ 1.0412\ 0.6941\ 0.6941\ 0.3471\ 0.6941\ 0.6941\ 0.3471\ 0.3471\ 0.3471\ 0.6941]
\]  

(6.9)

If, for each received value we test all possible three-bit combinations and compute the mean squared error using the following equation we can then apply the threshold to determine the possible states that would be present different stages of the trellis;

\[
\text{MSE} = (Y - X)^2
\]  

(6.10)

![Trellis for a sequence of 10 input bits.](image)
For a selected threshold value of $Th = 0.000006$ the trellis is shown in Fig 6.7. Note that since the optimum path through the trellis is the one indicated by black arrows we can take the middle bit of each selected state and the detected data will be:

$$[1 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]$$  \hspace{1cm} (6.11)

If we take the preamble out then the detected sequence will become:

$$[1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0]$$  \hspace{1cm} (6.12)
Chapter 7

SIMULATION RESULTS

7.1 Image Restoration Using 1D Wiener and Inverse Filters

Since convolution can be performed in the frequency domain and an image can be blurred by convolving it with a point spread function, it is simple to introduce blur into some perfect images to obtain blurred versions that can be used for simulation. Restoration methods can be tested on these blurred versions of the known images and quality of the results can be determined by simply comparing the original against the restored image. In this study tested images had sizes (20x20), (58x100) and (128x128). Performances were simulated for SNR values in the range 0-20dB, with 2 dB steps and the additive white gaussian noise in simulations was generated using related Matlab function. BER is calculated by computing the average between the total number of bits in error (the difference between the original bits and the estimated bits) over the total number of the original bits. Most of the simulations used through this chapter converged after one iteration except the simulation used for VA, VA-DF and modified trellis method. The relation between BER graphs and restored images is obvious; the curve which has the better performance mostly has the better restored image. For bit error rate (BER) computations a corrupted version of the original image was generated and used such that the signal to noise ratio would be:

\[
SNR = \frac{\text{variance}(g \ast h)}{\sigma_n^2}
\]

(7.1)

This is just the ratio of the noiseless image variance to the noise variance. In practice one usually would not have enough knowledge about the nature of the received noise signals. We therefore may consider \( K \) to be constant as previously shown in equation (3.16). Figure 7.1 below shows the performance of Wiener and Inverse filters at \( K \) values of 0.1, 0.01, and 0.001. From the first glance, one can note that if the value of \( K \) is increased, the performance would also increase. Particularly at high SNR the improvement increases rapidly. In the simulation related to Fig 7.1, a one-dimensional blur has been used to represent the effect between neighbor bits recorded in a one dimension such as the recorded bits in a track of magnetic tape.
Fig 7.1. Comparison of BER performance for Wiener and Inverse filters at K-values of 0.1, 0.01, 0.001 and 0. $h = [1/3 \ 1/3 \ 1/3]$.

An important point to note is that the inverse filter delivers the same performance for all the values of K selected for the Wiener filter. It does so because the estimated response using Inverse filtering doesn't depend on the constant K.

For the performance curves depicted in Fig 7.1, the selected point spread function is:

$$h(0, j) = \frac{1}{3}, \quad j = 0,1,2$$  \hspace{1cm} (7.2)

and the text image used in this simulation is the one shown in Fig 7.2. In this study a second set of comparisons were taken by trying to restore the $(128 \times 128)$ text image corrupted by the same point spread function. Figure 7.2 below illustrates the performance of Wiener and Inverse filters for a selected signal to noise ratio of 18dB. Wiener filter estimation at $K= 0.1$ is the most readable output when compared with the others. It was also observed that for a $K$-value of 0.001 the estimated object by using Wiener filter looks almost identical to the one, which has been restored by Inverse-filtering. Both outputs are unreadable. Also if one tries to further increase the value of $K$, only slight improvements will be achieved.
As discussed in Chapter 3, if one changes the variable $\gamma$, Wiener filter will be called Parametric Wiener filter. The simulation in Fig 7.3 has been run for $\gamma = 1.0, 0.1$ and $0.01$. It is obvious from the figure that as the $\gamma$ value gets larger, the performance of the Wiener-filter improves. Comparing results of Fig 7.1 and Fig 7.3 we note that for both the 1D-Wiener and its parametric version the performance will increase as $K$ or $[\gamma S_u/S_\gamma]$ increases. However the performance of the Inverse filter does not depend on the variable $\gamma$. In this work we also looked at the effect of increasing the PSF size on the general performance. Figure 7.4 indicates the attained performance curves for a $(1 \times 5)$ PSF like the one shown below:

$$h(0,j) = \frac{1}{5}, \quad j = 0,1, \ldots, 4 \quad (7.3)$$
Fig 7.3. Performance comparison between the Parametric Wiener filter at $\gamma=1.0, 0.1, 0.01$ and the inverse filter given the blur function $h = [1/3 \ 1/3 \ 1/3]$.

Fig 7.4. Effect of increasing PSF size on the performance of Wiener and Inverse filters [ $h(0,j) = [1/3], j = 0,1,2$ and $h(0,j) = [1/5], j=0, 1, ..., 4$ ]
In the (1×5) blur each bit has been corrupted by the effect of five neighboring bits in the same track. It is obvious that as the size of the 1D-PSF increases the performance for both the WF and IF decreases.

7.2 Image Restoration Using 2D Wiener and Inverse Filters

The dispersion of the media and the write-and read-back channel neighboring bits affect each other and cause amplitude distortions and peak shifts. As an example, for any storage techniques with high track densities one can observe a cross-talk between the tracks. This cross-talk is similar to a two-dimensional blur. In the following simulations different types of 2D blurs have been used. Figure 7.5 depicted below is for the following low pass PSFs;

\[
\begin{bmatrix}
0.01 & 0.07 & 0.01 \\
0.07 & 0.68 & 0.07 \\
0.01 & 0.07 & 0.01
\end{bmatrix}
\quad h_1
\]

\[
\begin{bmatrix}
0.01 & 0.12 & 0.01 \\
0.12 & 0.48 & 0.12 \\
0.01 & 0.12 & 0.01
\end{bmatrix}
\quad h_2
\]

\[ (7.4) \]

It can be seen from the BER plots that the second mask \( h_1 \), offers better performance than the first one, \( h_2 \). The reason is obvious from the matrix elements. The first mask boosts up each pixel in the corrupted image greater than the second mask. This is because the central pixel of the first mask has value greater than the central pixel of the second one. Another point to observe is that when the first mask is used, as SNR increases the Inverse filter’s performance becomes closer to that of the Wiener filter, particularly for SNR more than 14dB. Fig 7.6 shown below illustrates this result. These masks pass only the low frequency components. In general, a lowpass mask is responsible for the slow varying characteristics as well as eliminating high frequency components. From the restored images depicted in Fig 7.6 at SNR value of 15 dB, one can note that the restorations achieved by WF and IF when the first mask \( h_1 \) has been used are similar to the performance of wiener filter for second mask \( h_2 \).
Fig 7.5. Performance comparison between WF and IF while using two different lowpass masks $h_1$ and $h_2$. Central pixel of $h_1$ has a higher intensity value as shown in (7.4).

Fig 7.6. Text-image reconstructions using WF and IF for images blurred by lowpass masks $h_1$ and $h_2$. Restorations were carried out at SNR value of 15dB.
Fig 7.7. Performance curves for Wiener and Inverse filtering of an image corrupted by highpass masks \( h_1 \), \( h_2 \) and \( h_3 \) given in (7.4).

The BER graph depicted in Fig 7.7 illustrates the performance comparison between three types of high-boost filtering. High pass filters or high-boost filtering attenuate or eliminate low-frequency components and leave the high frequency components untouched. The object of these filtering is to characterize edges and other sharp details in an image. The masks used for these purposes were as follows:

\[
\begin{align*}
    h_1 &= \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 26 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\
    h_2 &= \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 17 & -1 \\ -1 & -1 & -1 \end{bmatrix} \\
    h_3 &= \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{bmatrix}
\end{align*}
\]

(7.4)

It can be observed from the BER graph that high-boost filters offer better results as the central pixel value increases. This is because the ability of each mask to pass high frequency components increases. From the first glance, one can observe that the performance of Inverse filter is better than that of Wiener filter at low central pixel values. Note that also the performances of Wiener and Inverse filters become the same when the first mask has been used or when the central pixel increases to a higher
value. There is a known problem introduced when this type of filtering is used. The problem, as discussed in Chapter 2, is that the Inverse filter fails to estimate PSF of the channel because of the degraded image has some zeros and the ratio one over zero is an undefined quantity. One could observe this problem when the sum of the negative values around a particular pixel in a degraded image equals to the central pixel value.

Figure 7.8. shows reconstructed versions of the aza.jpeg image that has in turn been corrupted by using the highpass filtering masks $h_1$, $h_2$, and $h_3$ given below:

$$
\begin{align*}
    h_1 &= \begin{bmatrix}
        -1 & -1 & -1 \\
        -1 & 8 & -1 \\
        -1 & -1 & -1 
    \end{bmatrix} &
    h_2 &= \begin{bmatrix}
        -1 & -1 & -1 \\
        -1 & 8.4 & -1 \\
        -1 & -1 & -1 
    \end{bmatrix} &
    h_3 &= \begin{bmatrix}
        -1 & -1 & -1 \\
        -1 & 12 & -1 \\
        -1 & -1 & -1 
    \end{bmatrix}
\end{align*}
$$

The experiment was observed at a SNR value of 12dB. For the first mask we note that Inverse filter is incapable and fails to restore the corrupted object due to the zero problem and hence in Fig 7.8 no output is shown. An interesting point when one compares the performance obtained from Wiener filter and Inverse filter is that, as the value of the central pixel increases the performance of both Wiener and Inverse filters increases and restoration of the blurred image improves. Also one can note that the edges of the blurred object are improved and enhanced. It is clear that images in Fig 7.8. have black backgrounds because the frequency components giving the contrast of the image have been eliminated by the highpass filters. In another meaning, usually the background of the image is white (its intensity values equal ones) and when these values are convolved with a highpass mask (sum of its value equal zero) the output will be all zeros or very small values therefore, the background of the image becomes black.
Fig 7.8. Text-image restorations for Wiener and Inverse filtering of an image blurred by highpass masks $h_1$, $h_2$, and $h_3$.

Similarly an original Einstein picture as the one depicted in Fig 7.9-(a) was corrupted with additive white gaussian noise (SNR= 8dB) and the highpass filtering mask indicated below:

$$h_2 = \begin{bmatrix}-1 & -1 & -1 \\ -1 & 8.4 & -1 \\ -1 & -1 & -1\end{bmatrix}$$  \hspace{1cm} (7.6)
The degraded version of the image is shown in Fig 7.9-(b) and fig 7.9-(c) shows the result of Wiener filtering which looks like the restored copy in Fig 7.9-(d) by Inverse-filtering. The output from Wiener filter is nearly the same with the Inverse filter output. This is because Wiener and Inverse filters nearly deliver same performance at low SNR values, as can be seen from Fig 7.7.

7.3 Novel Technique Based on Two Wiener Filters and Two Threshold Values

The usefulness of the novel de-blurring technique is demonstrated by comparing it with the previously suggested methods in the literature. The comparisons were based on bit error rate performances and text-image restorations for bi-level images of size (58×100). Simulations conducted were for vertical blur $h_1$, origin symmetric blur $h_2$, and Gaussian type mask $h_3$ indicated below:

$$ h_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad h_2 = \begin{bmatrix} 0.01 & 0.07 & 0.01 \\ 0.07 & 0.68 & 0.07 \\ 0.01 & 0.07 & 0.01 \end{bmatrix} \quad h_3 = \begin{bmatrix} 0.025 & 0.107 & 0.025 \\ 0.107 & 0.466 & 0.107 \\ 0.025 & 0.107 & 0.025 \end{bmatrix} $$ (7.7)

In this study the first and second thresholds used were 0.5 and 0.5 respectively and SNR was taken as:

$$ SNR = \frac{\sum_{i,j} h^2(i, j)}{\sigma_n^2} $$ (7.8)

It can be observed from Fig 7.10. that, the BER performance obtained using the novel method is better in the 0-12 dB range.
If the origin symmetric blur, \( h_2 \), is used, the new algorithm will deliver the best performance in the 0-14 dB range, as shown in Fig 7.11. Similarly when the Gaussian mask, \( h_3 \), is used for artificially blurring the image the new algorithm once again delivers the best BER performance in the 0-13.5dB range (Fig 7.12).
Fig 7.13. Comparison of BER performance between known methods and the novel restoration algorithm using the EMU.jpeg image degraded by the Gaussian blur $h_3$.

The second best performance in all three cases is provided by the iterative parallel data detection algorithm, which delivers better performance at higher signal to noise ratios. The text-image restoration carried out for the Gaussian blur is shown in Fig. 7.13. It can be observed that the most noise free restored image is obtained using our suggested technique. However there are some pixels missing from the letters. This problem can be solved by applying a closing operation that is equivalent to a dilation followed by erosion that uses the same structuring element.

Fig 7.13. Restoring the image EMU.jpeg corrupted by Gaussian blur. Image size: (58×100), experiment observed at SNR=10dB.
7.4 Viterbi Algorithm (VA) Used for De-blurring Degraded Images

To evaluate the performance of the Viterbi algorithm, we simulated the detection process for origin symmetric mask $h_1$, origin symmetric worst-case mask $h_2$, and gaussian blur $h_3$ respectively:

$$
\begin{align*}
    h_1 &= \begin{bmatrix} 0.01 & 0.07 & 0.01 \\ 0.07 & 0.68 & 0.07 \\ 0.01 & 0.07 & 0.01 \end{bmatrix} \\
    h_2 &= \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \\
    h_3 &= \begin{bmatrix} 0.025 & 0.107 & 0.025 \\ 0.107 & 0.466 & 0.107 \\ 0.025 & 0.107 & 0.025 \end{bmatrix}
\end{align*}
$$

(7.9)

From Fig 7.14 we can observe that, VA with decision feedback offers the best performance, at SNRs higher than (8dB) that is due to the assumption of the row above the current one is known. One can note that VA at low SNRs delivers better BER value than VA-DF because the probability of the first rows in VA-DF to be detected correctly at low SNRs is low. This performance is followed by the novel method described in section-7.3. The computational complicity of the VA is a disadvantage, while the other techniques converge after one or two iteration. With the origin symmetric worst-case mask, Wiener filter delivers the best performance for the full range of SNR and all other techniques including the VA-DF deliver poor performance as shown in Fig 7.15.

Fig 7.14. Comparison of BER performance for linear and nonlinear restoration algorithms using the EMU.jpeg image degraded by the origin symmetric PSF $h_1$. 

Image size: 58×100
Image: EMU.jpeg
The power of the VA-DF algorithm in restoring degraded bi-level text images can easily be seen from Fig 7.16, below. Among the restored images VA-DF restored one is the best and the novel method of section 7.3 is second best in term of the noise. For this experiment the original image was blurred by the origin symmetric mask, $h_1$, and the de-blurring process was carried out at an SNR value of 12 dB.

Fig 7.16. Text image restoration of the image EMU.jpeg corrupted by origin symmetric blur $h_1$. Image size: (58×100), experiment observed at SNR=12dB.
Fig 7.17. Comparison of BER performance for linear and nonlinear restoration algorithms using the EMU.jpeg image degraded by the Gaussian mask $h_3$.

The bad performance indicated by VA-DF method when the worst case origin symmetric mask is used can be explained with the fact that when the image is exceedingly blurred the possibility of iterative decision feedback leading to error propagation quicker is a lot higher than before. Finally, from Fig.7.17, one can observe that the VA method delivers the best performance for SNRs lower than 8dB when a Gaussian PSF is used as the blurring mask.

7.5 Novel One Dimensional De-Blurring Algorithm Based on Minimum Mean Squared Error and A Selected Fixed Threshold

While simulating the BER performance using the second novel technique the following point-spread function was used:

$$h = \{1/3 \ 1/3 \ 1/3\}$$  \hspace{1cm} (7.10)

The input data had size (20×20) and the information frame was partitioned into stripes of three bits long as described in Chapter 6. The plot shown in Fig 7.18 below,
compares the performances of various algorithms that have been previously discussed with that of the new method in terms of the bit-error-rate versus signal to noise ratio.

As could be observed from the plot the new 1D de-blurring method performs better than the Inverse and Wiener filtering techniques. It is also apparent that both the new proposed technique and the threshold based filtering approach perform better than the iterative parallel detection method for signal to noise ratios that are considered low (0-9 dB). For SNR values at or exceeding 13 dBs the new method performs better than the threshold method however for values exceeding 9 dBs the iterative parallel detection method delivers a better performance rate than the new method.

The novel trellis method was also simulated over various-length quasi static Rayleigh fading channels providing the results shown in Fig 7.19. The new algorithm delivers a better performance than the conventional hard and soft decisions VA decoding for a convolutional code of rate $\frac{1}{2}$ and constraint length $K=3$ at SNR values exceeding 15dB. Maximum number of paths simulated was 4.
Fig 7.19. Performance comparisons between new trellis method and conventional hard & soft detection VA, K=3, r =1/2 over various length quasi static Rayleigh fading channels.

The time varying multipath channel used herein is a frequency nonselective channel. The tap coefficients in the tapped delay line model were characterized as complex-valued Gaussian random processes. Each tap is presented as,

\[ c(t) = v(t)e^{j\phi(t)} \]  
[7.11]

\[ v(t) = \sqrt{c_r^2(t) + c_i^2(t)} \]  
[7.12]

where \( c_r(t) \) and \( c_i(t) \) represent real-valued Gaussian random processes with zero mean. And \( \phi(t) \) is uniformly distributed over the interval [0,2\( \pi \)). The lowpass received signal is,

\[ r_n(t) = \sum_n v_n(t)e^{j\phi_n(t)}s_n(t - \tau_n) + \text{AWGN} \]  
[7.13]

where \( \tau_n \) is the propagation delay for the \( n \)-path.
Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

The investigations that were carried out throughout this thesis started with simulations of the Inverse filter that was proposed in the early 1960s and continued till the submission of the Iterative Parallel Data Detection technique proposed by Neifeld and Chugg in 1996. Later two new de-blurring algorithms were proposed and simulations were conducted for comparing the performances attained using these novel methods with the ones for previously described techniques. It has been observed that both new methods would provide moderate gains.

When an Inverse filter is used for the restoration of a corrupted image the disadvantage is that the noise and the actual data are enhanced together. The Wiener filter does consider this point and hence it would deliver better performance than the former. As described in Chapter-3 when the noise is unknown the ratio of the power spectral densities of the received signal and the noise can be taken as constant. It was observed that as this ratio gets larger the performance of the Wiener filter would tend to increase as depicted in Fig 7.2. It has also been observed that when the PSF has equal pixel values both the Inverse and Wiener filters would perform poorly. Finally we can state that these linear filter based techniques are non-optimal since they do not make use of the a priori knowledge hidden in the original data for restoration process.

The 1D-VA, 2D-VA and VA with decision feedback, which were also used throughout this thesis as non-linear filtering techniques usually, would offer better performance in comparison to linear filtering techniques. However such techniques could at times suffer from severe blur types such as the worst-case origin symmetric PSFs. Even though the 2D-VA described would deliver better performance it would also suffer from computational complexity. The computational complexity would increases as the
size of the corrupted image increases or if the blur size increases. VA-DF is an alternative technique, which takes less restoration time and is only possible at the expense of the memory required for storing rows of previously detected data. In a system with a $(3 \times 3)$ transfer function, one row of data must be retained in order to estimate the subsequent row. However these disadvantages are not enough to eliminate the BER gains achieved when feedback is incorporated.

Iterative Data Detection technique has the advantage that from the second iteration during the computer simulation the result converges and delivers better performance than both Wiener and Inverse filters. The amount of gains possible with this method depend on the output of the Wiener filter. Whether good or bad the iterative approach tries to enhance these results. Finally it has been noted that the Iterative Data Detection still performs poorer than the 2D-VA.

The double-threshold novel technique, for most blur types, offers better BER performance than other well established techniques with the exception of the 2D-VA. However with severe blur types such as $h(i,j) = 1/9$, the new method could perform better than the VA. Finally, the second novel technique, which we refer to as the “modified trellis approach” performs poorly at low signal to noise ratios but it is very powerful at mid range SNR values as given in Fig 7.19.

8.2 Future work

It is clear that all previous methods are still none optimal. If one tries to improve the performance then the computational complexity of the VA based non-linear algorithms will increase. Our proposed ‘modified trellis search’ algorithm so far has assumed that for a $(3 \times 3)$ 2D block processing only the first row of data is correct. This would mean $16$ different possible 4-bit states (i.e. 0000) at each step of the trellis. However if we scan not only the first but also the second row by 1D-VA and assume that the first two detected rows are known and correct then the total number of states will drop to four, a complexity reduction by a factor of four. This can be one idea to investigate to see the BER possible at this reduced complexity.
Secondly the data frame in the modified trellis approach can be encoded by any one of the known coding methods before striping and sending it to the channel. This should deliver somewhat better BERs.

Thirdly it is now possible to extend this work from bi-level images to gray-level images. For example if we assume 8-bit gray level images we would have intensity values between 0-255 representing the data page. We can initially convert each intensity value to its binary equivalent and concatenate these bits to form a data frame just as the one used in the modified trellis approach of Chapter-6. The data frame can then be broken into stripes and the processing is as before. After detection we again will have a frame of binary digits. At this point we can break this frame into 8-bit long sequences and convert each sequence to decimal. These decimal values would represent the de-blurred gray level image intensity values. The published work so far only talks about bi-level images and not the gray-level ones. Our modified trellis algorithm makes it very easy to do this extension and hence it will be great to continue in this direction as future work.

Finally since this last approach has a parallel architecture it is also possible to try to implement the algorithm using DSPs and develop an actual real time system. This system will be a lot faster because of its parallel structure however it may need to use more hardware.
REFERENCES


APPENDIX-A
The appendix contains copies of the two national and two international conference papers that have been accepted (2000-2001 period).