

8-1-2005

# Optimal image deconvolution by range and noise moment constraints

Angelica Maria Diaz  
*Florida International University*

**DOI:** 10.25148/etd.FI14062269

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

 Part of the [Electrical and Computer Engineering Commons](#)

---

## Recommended Citation

Diaz, Angelica Maria, "Optimal image deconvolution by range and noise moment constraints" (2005). *FIU Electronic Theses and Dissertations*. 2799.

<https://digitalcommons.fiu.edu/etd/2799>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact [dcc@fiu.edu](mailto:dcc@fiu.edu).

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

OPTIMAL IMAGE DECONVOLUTION BY RANGE AND NOISE MOMENT  
CONSTRAINTS

A thesis submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Angelica Maria Diaz

2005

To: Dean Vish Prasad  
College of Engineering and Computing

This thesis, written by Angelica Maria Diaz, and entitled Optimal Image Deconvolution by Range and Noise Moment Constraints, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this thesis and recommend that it be approved.

Jean Andrian

Armando Barreto

Frank M. Candocia, Major Professor

Date of Defense: August 1, 2005

The thesis of Angelica Maria Diaz is approved.

Dean Vish Prasad  
College of Engineering and Computing

Dean Douglas Wartzok  
University Graduate School

Florida International University, 2005

## DEDICATION

This thesis is dedicated to the love of my life, Jong, my parents, Danny and Alicia, my brother and sisters, and to my dearest Isabella and Sebastian.

## ACKNOWLEDGMENTS

I would like to express my sincere gratitude for everyone that helped and supported me through all the years I have spent in the Department of Electrical Engineering. In particular, I would like to thank Dr. Frank Candocia. This thesis would not have been possible without his guidance and support. To my committee members Dr. Jean Andrian and Dr. Armando Barreto, I thank them for their input and advice on this work.

I would also like to thank my fellow researchers at the Image Understanding Lab for making these years memorable and so enriching.

# ABSTRACT OF THE THESIS

## OPTIMAL IMAGE DECONVOLUTION BY RANGE AND NOISE MOMENT

### CONSTRAINTS

by

Angelica Maria Diaz

Florida International University, 2005

Miami, Florida

Professor Frank M. Candocia, Major Professor

Image deconvolution, also known as image restoration, is concerned with the estimation of an uncorrupted image from a noisy, degraded one. The degradation of this image may be caused by defects of optical lenses, nonlinearity of the electro-optical sensor, relative motion between an object and camera, wrong focus, etc. By assuming a degradation model, one can formulate and develop a restoration algorithm. In this thesis, the developed algorithms are iterative deconvolution methods based on noise moment and pixel range constraints. The moments were used to ensure that noise associated with the deconvolution solution satisfies predetermined statistics. The pixel range constraints were also used to ensure the solution is within predetermined pixel value bounds. This addresses the critical issue of noise amplification at those frequencies where the point-spread function (the blurring function) contains frequency nulls. The solution's dependence on the number of moments is examined and the performance of the deconvolution approach is compared with existing and well established deconvolution methods such as Wiener filtering and inverse filtering.

## TABLE OF CONTENTS

CHAPTER	PAGE
1. Introduction.....	1
2. Existing Classes of Image Deconvolution Techniques.....	4
2.1 Inverse Filtering.....	4
2.2 Least-squares Filtering.....	4
2.3 Iterative Filters.....	6
2.3.1 Iterative Constrained Deconvolution.....	6
2.3.2. CLEAN.....	7
2.4 Wavelet-Based Deconvolution.....	8
3. Standard Deconvolution Approaches.....	10
3.1 Convolution Models.....	10
3.1.1 Infinite Extent Discrete Convolution.....	11
3.1.2 Finite Extent Discrete Convolution.....	12
3.2 Deconvolution by Filtering.....	16
3.2.1 The Inverse Filter.....	17
3.2.2 The Wiener Filter.....	25
3.3 Implementation Notes.....	27
4. Noise and Range Constrained Deconvolution.....	29
4.1 Range Constraint.....	30
4.2 Noise Moment Constraints.....	30
4.3 The Probability Weighted Moments.....	32
4.3.1 Uniformly Distributed PW Moments.....	33
4.3.2 Gaussian Distributed PW Moments.....	34
4.4 Estimating Probability Weighted Moments.....	36
4.5 Performing the Deconvolution.....	37
4.6 Optimization Using Standard Moment Constraints.....	39
4.7 Optimization Using Probability Weighted Moment Constraints.....	42
5. Experimental Results.....	49
5.1 Filters Considered.....	49
5.2 Analysis and Comparison.....	56
5.2.1 Quantitative Measures Used.....	56
5.2.2 Performance of Methods.....	58
6. Conclusion and Future Work.....	92
List of References.....	94
Appendices.....	96

## LIST OF FIGURES

FIGURE	PAGE
3-1 Graphical illustration of the two dimensional convolution between the finite extent $x$ and “sliding window” $h$ .....	14
5-1 Spectral response of the “well behaved” low pass filter given by eqn. (5.4) .....	51
5-2 Spectral response of the second “well-behaved” low pass filter given by eqn. (5.5) .....	53
5-3 Spectral response of the “poorly” behaved low pass filter given by eqn. (5.6) .....	54
5-4 Spectral response of the “poorly” behaved low pass filter given by eqn. (5.7) .....	55
5-5 Results with the 3-tap “well-behaved” filter given by eqn. (5.4) .....	64
5-6 Restoration results corresponding to the plot in fig. (5-5a) when SNR=5dB and $M=1$ .....	65
5-7 Restoration results corresponding to the plot in fig. (5-5a) when SNR=15dB and $M=1$ .....	66
5-8 Restoration results corresponding to the plot in fig. (5-5a) when SNR=25dB and $M=1$ .....	67
5-9 Restoration results corresponding to the plot in fig. (5-5b) when SNR=5dB and $M=3$ .....	68
5-10 Restoration results corresponding to the plot in fig. (5-5b) when SNR=15dB and $M=3$ .....	69
5-11 Restoration results corresponding to the plot in fig. (5-5b) when SNR=25dB and $M=3$ .....	70
5-12 Results with the 5-tap “well-behaved” filter given by eqn. (5.5) .....	71
5-13 Restoration results corresponding to the plot in fig. (5-12a) when SNR=15dB and $M=1$ .....	72

5-14 Restoration results corresponding to the plot in fig. (5-12a) when SNR=25dB and $M=1$ .....	73
5-15 Restoration results corresponding to the plot in fig. (5-12a) when SNR=35dB and $M=1$ .....	74
5-16 Restoration results corresponding to the plot in fig. (5-12b) when SNR=15dB and $M=3$ .....	75
5-17 Restoration results corresponding to the plot in fig. (5-12b) when SNR=25dB and $M=3$ .....	76
5-18 Restoration results corresponding to the plot in fig. (5-12b) when SNR=35dB and $M=3$ .....	77
5-19 Results with the 3-tap “poorly behaved” filter given by eqn. (5.6).....	78
5-20 Restoration results corresponding to the plot in fig. (5-19a) when SNR=15dB and $M=1$ .....	79
5-21 Restoration results corresponding to the plot in fig. (5-19a) when SNR=25dB and $M=1$ .....	80
5-22 Restoration results corresponding to the plot in fig. (5-19a) when SNR=35dB and $M=1$ .....	81
5-23 Restoration results corresponding to the plot in fig. (5-19b) when SNR=15dB and $M=3$ .....	82
5-24 Restoration results corresponding to the plot in fig. (5-19b) when SNR=25dB and $M=3$ .....	83
5-25 Restoration results corresponding to the plot in fig. (5-19b) when SNR=35dB and $M=3$ .....	84
5-26 Results with the 5-tap “poorly behaved” filter given by eqn. (5.7).....	85
5-27 Restoration results corresponding to the plot in fig. (5-26a) when SNR=15dB and $M=1$ .....	86
5-28 Restoration results corresponding to the plot in fig. (5-26a) when SNR=25dB and $M=1$ .....	87
5-29 Restoration results corresponding to the plot in fig. (5-26a) when SNR=35dB and $M=1$ .....	88

5-30	Restoration results corresponding to the plot in fig. (5-26b) when SNR=15dB and $M=3$ .....	89
5-31	Restoration results corresponding to the plot in fig. (5-26b) when SNR=25dB and $M=3$ .....	90
5-32	Restoration results corresponding to the plot in fig. (5-26b) when SNR=35dB and $M=3$ .....	91

## Chapter 1

### Introduction

Advances in digital image restoration first came about in the 1950s and early 1960s by scientists involved in the space programs of the United States and the former Soviet Union [Kats91]. Many of the space programs were interested in capturing images of the Earth and the solar system. However, the images captured during the planetary missions experienced many photogenic degradations. The loss of information due to image degradation was devastating. The necessity to retrieve the information lost drove the engineering community to work on algorithms that would retrieve meaningful information from the degraded images.

Digital image restoration not only plays an important role in astronomical imaging but also in the medical field, where restoration has been used for filtering of Poisson distributed film-grain noise in chest X-rays, mammograms, etc. [Kats91]. It has been also used by law enforcement agencies to restore blurry photographs of license plates and crime scenes which may provide the only link to solving a crime. Digital image restoration is by no means limited to the previously listed examples as there are a wide variety of applications that can benefit from such processing. In particular any image obtained from a process or sensor that is degraded due to defects in optical lenses, nonlinearity of sensors, relative motion between object and camera, wrong focus, etc. is a good candidate for restoration.

In the field of digital image restoration, methods are developed to recover an original image from the degraded observation. These methods perform the restoration

task by assuming models, not only for the cause of the degradation and/or noise, but also for the original image. They also make use of regularization theory since the problem related to image restoration is an ill-posed one. Regularization theory involves the use of constraints related to the type of signal being dealt with and the manner in which the constraints affect the effectiveness and the efficiency of the deconvolution algorithm [Star02]. In this thesis, we examine how image deconvolution can be improved, in particular, as it relates to the “shaping” of noise in the solution.

Chapter 2 provides a review of previous work done in the field of image restoration so as to better identify where the contributions of this thesis lie.

Chapter 3 provides a description to standard approaches in deconvolution. It details the convolution and the degradation model used in this work. It also provides a mathematical description of the Wiener and inverse filtering approaches. These approaches will be used as a basis for comparison with the algorithm developed in this thesis.

Chapter 4 provides the mathematical description of the deconvolution approaches we have developed. These algorithms are iterative methods based on noise moment and pixel range constraints. They also address issues common to image restoration such as establishing an objective stopping criteria for iterative deconvolution methods.

Chapter 5 contains experimental results obtained from various image restoration algorithms. It provides a quantitative and qualitative analysis of the algorithms developed in this thesis relative to the standard approaches mentioned in chapter 3.

Finally, chapter 6 concludes with a summary of what was accomplished. We also take the opportunity to address future research issues in image restoration and deconvolution.

## Chapter 2

### Existing Classes of Image Deconvolution Techniques

The majority of the techniques used in image restoration try to model the degradation and/or noise and then apply an inverse procedure to obtain an approximation of the original image. Since image restoration is an ill-posed problem most of these techniques use regularization theory to yield meaningful answers and approximations to the given ill-posed problem. This regularization theory involves the use of constraints related to the type of signal we are dealing with and the way we use these constraints affects the effectiveness and the efficiency of the deconvolution algorithm [Star02].

Examples of commonly used constraints include nonnegativity, which means that all pixel values in the estimate resulting from the deconvolution process must be positive. Other types of constraints include boundary constraints and deterministic constraints. These help to reduce the set of feasible solutions [Axel00]. In addition, constraints can also be placed on the noise that is an intrinsic part of all measured images. The presence of noise is one critical factor that makes the deconvolution process particularly difficult. In this chapter we will describe some of the most used methods for performing image deconvolution.

#### 2.1 Inverse Filtering

An inverse filter is typically a linear filter whose point spread function (PSF) is the inverse of the blurring function. The implementation of the linear filter in the spatial domain can be difficult. In contrast, the spectral counterpart is usually easier. The advantage of the inverse filtering technique is that it requires only the blur PSF as a priori

knowledge. This technique is computationally very quick and allows for the perfect restoration of an image if (1) the PSF has no frequency nulls, i.e., frequency components at zero spectral magnitude and (2) there is no noise in the image. When noise is present, its effects are necessarily amplified by this technique. An image's spatial frequencies eliminated by the filter's nulls cannot be recovered in this technique. In fact, no inverse filter will exist for performing this filtering although an approximation to it can be employed.

## 2.2 Least-squares Filtering

To overcome the noise sensitivity of the inverse filter, a number of restoration filters have been developed that are collectively called least-squares filters. The two most commonly used filters are the Wiener filter and the constrained least-squares filter.

The Wiener filter is a linear, spatially invariant filter in which the point spread function is chosen such that it minimizes the mean-squared error (MSE) between the ideal and restored images. The solution to this minimization problem is known as the Wiener filter. It is typically defined in the spectral domain. In the typical situation where an image is noisy, this approach trades-off the restoration by inverse filtering and suppression of noise for those frequencies where the PSF is close to or equal to zero. In chapter 3, we will describe in more detail this particular approach [Lage00].

The constrained least-squares filter is another approach that also overcomes the problem that the inverse filter has, i.e. excessive noise amplification, by using the deterministic prior information about the original image as a regularization parameter [Lage00].

## **2.3 Iterative Filters**

Inverse and least-squares filters usually result from the closed form solution of a formulated problem. Iterative filtering, however, is an approach where small steps are taken towards the restored image and this is usually done in the spatial domain. This is in contrast to the “one-step”, closed form solution obtained in the inverse and Wiener filters. The effect is to approach the solution and essentially stop before noise amplification “takes over”, i.e. the iterations can be terminated whenever an acceptable restoration result is achieved. Any iteration past this point will result in increasingly noisy and visually degraded images. To more intelligently and mathematically produce a stopping criteria, constraints on the restored image or some other quantity must be introduced to the iterative process. These filters effectively seek to trade-off accuracy in restoration with noise amplification. This is further addressed in the next two sections. But before doing this, we note that all iterative deconvolution begins with an initial guess at the restored image. This initial estimate starts the iteration process. Due to typical lack of a priori knowledge, it is very common to use the given degraded image as the initial guess. In these iterative schemes, the correction per iteration is based on the residual between the observed image and the blurred image estimate. Iterative methods will be terminated after a finite number of iterations to obtain an approximation of the true image.

### **2.3.1 Iterative Constrained Deconvolution**

Because the performance of iterative type approaches is compromised by their sensitivity to noise in the signal or to error in the estimate of the point spread function, adding constraints is necessary. The typical constraints used include clamping the values

of the restored image so that they lie within a known reasonable dynamic range or limiting the amount of adjustment that can occur at any pixel. These limits may be imposed by setting thresholds or by introducing nonlinear range-limiting functions. Other useful constraints include those that restrict the possible noise space of the final solution. If these constraints are not present, typical amplification of noise results. Lack of a priori noise statistics might preclude the reasonable use of this type of constraint. This is the reason why many approaches look for a linear alternative to performing the restoration.

An approach that can be included in this category is the one proposed by [Hare00] in which the previous image estimate is pre-filtered using a stabilizing function that is updated based on current error and noise estimates. This approach tries to diminish the noise by the use of a second regularizing operator resulting in a hybrid iteration technique.

### **2.3.2. CLEAN**

The CLEAN method is a nonlinear iterative algorithm that assumes that the image is composed of point sources. In each iteration, this algorithm finds the strength and location of the brightest point of the image. It then subtracts the PSF degraded image (also called the dirty beam) multiplied by the peak strength and gain factor at this point, from the image (called the dirty map) at this location. This resulting map is then used in a repeated process until a prespecified limit is reached. After each iteration, what is left are the residuals, which are added to the convolution of a clean beam (or the so called ideal PSF) and point sources to yield the restored image or clean map. This algorithm is largely

used in radio astronomy but because of its computational complexity and basis of point sources, is not used in many other applications [Star02].

## 2.4 Wavelet-Based Deconvolution

The regularization methods of this section make use of the wavelet transform. This transformation's use has increased because of its ability to represent sharp discontinuities. Some of these related algorithms propose an inverse estimation procedure which combines Fourier analysis with wavelet expansion [Neel99]. The proposed algorithms comprise Fourier-domain system inversion (a regularized inverse filter which allows the algorithm to work even when the system is non-invertible) followed by wavelet-domain noise suppression. This is done by using a mean square error metric to strike an optimal balance between Fourier-domain regularization (matched to the system) and wavelet-domain regularization (matched to the signal/image).

Another algorithm, introduced by [Figu03] proposed an expectation-maximization (EM) approach for image restoration based on a penalized likelihood which is formulated in the wavelet domain. In this algorithm, regularization is achieved by promoting a reconstruction with relative low-complexity that is expressed in terms of the wavelet coefficients and takes advantage of the sparsity capabilities of the wavelet representation. The problem, in general, with methods based on the wavelet representation is that most of them require very demanding optimization methods, since the convolution operator is generally quite difficult to represent in the wavelet domain. This naturally suggests the possibility of combining Fourier-based deconvolution and wavelet-based denoising. The method in [Jalo02] proposes a deconvolution algorithm based on the EM procedure.

Here, the deconvolution problem is addressed using only a denoising technique. This approach consists of alternating a Landweber step and a denoising step. The denoising step is performed using a complex wavelet transform, ensuring both translational and rotational invariance properties and a spatially adaptive prior model.

## Chapter 3

### Standard Deconvolution Approaches

This chapter will present and describe the general image degradation model and subsequently develop the convolutional model that will be used in this thesis. In addition, the two most standard deconvolution approaches of a direct inverse filtering and Wiener filtering are presented and explained as they will serve as the comparative basis for the results presented in chapter 5.

#### 3.1 Convolution Models

The purpose of this section is to establish the convolution model that will be used in this thesis. It is a finite-extent model that addresses the limited number of data that is ever only available in practice.

We begin by noting that, in general, a continuous degraded image can be represented by the following equation

$$y(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_c(t_1, t_2, \tau_1, \tau_2) x(\tau_1, \tau_2) d\tau_1 d\tau_2 + \eta(t_1, t_2) \quad (3.1)$$

where  $y(t_1, t_2)$  is the degraded image,  $x$  is the original image,  $h_c$  is a continuous and spatially varying transformation, and  $\eta(t_1, t_2)$  describes an additive noise random process. By assuming all image objects are in focus and that higher order lens effects are negligible, the degradation model above conforms to one of a spatially invariant filtering of the original imaged signal [Kats84]. In this case, the general degradation model simplifies to the two-dimensional convolution of the original image  $x$  and the filter function  $h_c$ , plus the noise  $\eta$ . This is expressed as follows:

$$y(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_c(t_1 - \tau_1, t_2 - \tau_2) x(\tau_1, \tau_2) d\tau_1 d\tau_2 + \eta(t_1, t_2) \quad (3.2)$$

### 3.1.1 Infinite Extent Discrete Convolution

In typical fashion, the discretized convolution model is obtained by sampling the continuous relationship just described. Specifically, with uniform sampling on the input, the model of equation (3.2) can be reduced to a discrete approximation of the form in equation (3.3) to

$$\begin{aligned} y(t_1, t_2) &= \lim_{T_x \rightarrow 0} \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h_c(t_1 - k_1 T_x, t_2 - k_2 T_x) x(k_1 T_x, k_2 T_x) T_x^2 + \eta(t_1, t_2) \\ &\approx \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} T_x^2 h_c(t_1 - k_1 T_x, t_2 - k_2 T_x) x(k_1 T_x, k_2 T_x) + \eta(t_1, t_2) \end{aligned} \quad (3.3)$$

where  $T_x$  is the sample spacing along each spatial axis used in approximating the convolution integral. If we also sample the “output” (or degraded image  $y$ ) along each spatial axis with spacing  $T_y$ , we obtain

$$y(n_1 T_y, n_2 T_y) \approx \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} T_x^2 h_c(n_1 T_y - k_1 T_x, n_2 T_y - k_2 T_x) x(k_1 T_x, k_2 T_x) + \eta(n_1 T_y, n_2 T_y) \quad (3.4)$$

Since it is common to assume that the “input” and “output” signals have the same sampling period, we will let  $T = T_x = T_y$ . Note that if  $T_x < T_y$ , we would be dealing with the more difficult problem of superresolution as the input samples would need to be determined at a finer sample spacing than those in the observed image  $y$ . So, combining  $h_c$  and  $T$  into an equivalent filter, we have

$$y(n_1 T, n_2 T) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h((n_1 - k_1)T, (n_2 - k_2)T) x(k_1 T, k_2 T) + \eta(n_1 T, n_2 T) \quad (3.5)$$

where  $h = T_x h_c$ . Eqn. (3.5) will equivalently be expressed as the sequence relation

$$\begin{aligned}
y[n_1, n_2] &= \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} h[n_1 - k_1, n_2 - k_2] x[k_1, k_2] + \eta[n_1, n_2] \\
&= h[n_1, n_2] * x[n_1, n_2] + \eta[n_1, n_2]
\end{aligned} \tag{3.6}$$

where  $*$  denotes convolution. In our notation, we will use square brackets  $[ ]$  to denote a sequence of samples and rounded brackets  $( )$  to denote a continuous function.

### 3.1.2 Finite Extent Discrete Convolution

When performing deconvolution, only a finite amount of observed, sampled data will be available. Therefore, we need to develop a finite-extent convolution model from the discretized one in eqn. (3.6) that is defined over a limited number of samples, i.e.  $N_1 \times N_2$  number of rows and columns of the observed image  $y$ . The result of this extent limiting is given by

$$y[n_1, n_2] = \sum_{k_1=n_1-K_1}^{n_1+K_1} \sum_{k_2=n_2-K_2}^{n_2+K_2} h[n_1 - k_1, n_2 - k_2] x[k_1, k_2] \tag{3.7}$$

for  $n_1 = 0, \dots, N_1 - 1$  and  $n_2 = 0, \dots, N_2 - 1$  and where

$$K_i = \frac{M_i - 1}{2} \tag{3.8}$$

In eqn. (3.8),  $M_i$  represents the number of samples in the region of support of the discrete point spread function  $h$  along dimension  $i$  where  $i=1,2$ . That is,  $h$  is an  $M_1 \times M_2$  sized finite impulse response filter. Without loss of generality, we will assume  $M_i$  is an odd number for this presentation.

Notice that  $h[m_1, m_2]$  is defined for  $m_1 = -K_1, \dots, K_1$  and  $m_2 = -K_2, \dots, K_2$  and is considered zero outside this support region. In this way,  $x[k_1, k_2]$  is defined over samples

$k_1 = -K_1, \dots, N_1 + K_1 - 1$  and  $k_2 = -K_2, \dots, N_2 + K_2 - 1$ . Figure 3-1 serves to graphically illustrate the two dimensional convolution between the finite extent  $x$  and  $h$  that results in the  $y$  of eqn. (3.7). Note that  $x$  has support size of  $(N_1 + M_1 - 1) \times (N_2 + M_2 - 1)$  samples,  $M_1$  and  $M_2$  are the support size (number of rows and columns) of  $h$ , and  $y$  has support size  $N_1 \times N_2$ . This definition is one where all observed points in  $y$  result from the point spread function's weighted sum on the original samples of  $x$ . In this finite extent model, we say  $y$  results from the 'valid' portion of the convolution between defined samples of  $x$  and  $h$ .

It is very convenient to express the finite extent convolution of eqn. (3.7) in matrix/vector form. This is particularly useful and compact when convolution-related derivatives must be found. This equivalent form of eqn. (3.7) is

$$\mathbf{y} = \mathbf{H} \mathbf{x} + \boldsymbol{\eta} \quad (3.9)$$

where  $\mathbf{H}$  is a  $N_1 N_2 \times (N_1 + M_1 - 1)(N_2 + M_2 - 1)$  matrix known as a convolution matrix and  $\mathbf{y}$ ,  $\mathbf{x}$  and  $\boldsymbol{\eta}$  are vectors containing the samples of their respective signals. To explain how to establish this matrix, we begin with a one dimensional version of eqn. (3.7). That is, consider the relation

$$y[n] = \sum_{k=n-K}^{n+K} h[n-k]x[k] \quad (3.10)$$

where  $n = 0, \dots, N-1$  and  $k = -K, \dots, N+K-1$ . From this, we see  $y$  is a sequence of  $N$  samples  $n = 0, \dots, N-1$ ,  $h$  is a sequence of  $M$  samples and  $x$  is a sequence of  $N+M-1$  samples. In matrix/vector form, we write

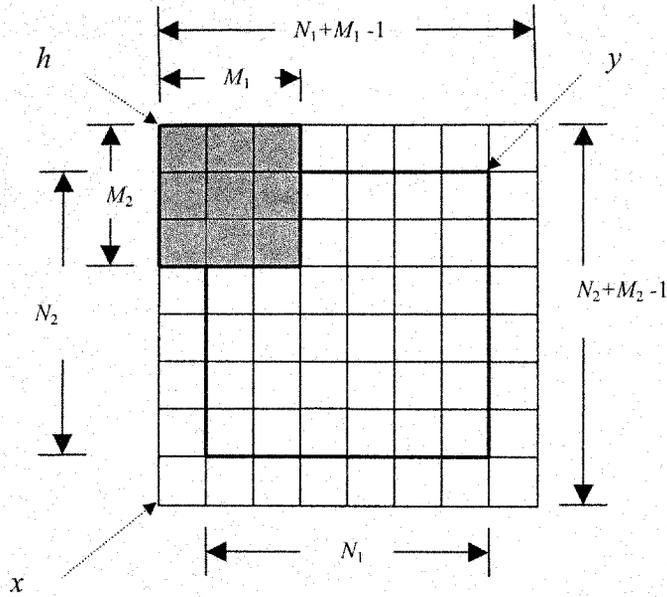


Figure 3-1. Graphical illustration of the two dimensional convolution between the finite extent  $x$  (which is represented by the big square) and “sliding window”  $h$  (which is represented by the grayed-in region). Notice that our finite extent convolution of “valid” pixels results in the smaller and bold square region that is  $y$ . In this example, we have used  $M_1 = M_2 = 3$ ,  $N_1 = N_2 = 6$  and this results in an  $x$  size of  $N_1 + M_1 - 1 = N_2 + M_2 - 1 = 8$ .

$$\begin{aligned}
 \mathbf{y} &= \underbrace{\begin{pmatrix} y[0] \\ y[1] \\ \vdots \\ y[N-1] \end{pmatrix}}_{N \text{ samples}} = \underbrace{\begin{pmatrix} h[+K] & \dots & h[-K] & 0 & \dots & 0 \\ 0 & h[+K] & \dots & h[-K] & & \\ \vdots & & \ddots & & \ddots & 0 \\ 0 & \dots & 0 & h[+K] & \dots & h[-K] \end{pmatrix}}_{N \times (N+M-1) \text{ samples}} \underbrace{\begin{pmatrix} x[-K] \\ x[-K+1] \\ \vdots \\ x[N+K-1] \end{pmatrix}}_{N+M-1 \text{ samples}} \\
 &= \mathbf{H} \cdot \mathbf{x}
 \end{aligned} \tag{3.11}$$

such that

$$\mathbf{H} = \left\{ \begin{array}{ll} h(n-k) & -K \leq n-k \leq K \\ 0 & \text{otherwise} \end{array} \right\} \left. \begin{array}{l} n=0, \dots, N-1 \\ k=-K, \dots, N+K-1 \end{array} \right\} \tag{3.12}$$

Note that the notation in eqn. (3.12) is a compact form of expressing the elements of a matrix. That is,  $\mathbf{Z} = \{z(n_1, n_2)\}_{n_1=0, \dots, N_1-1, n_2=0, \dots, N_2-1}$  would indicate that  $\mathbf{Z}$  is an  $N_1 \times N_2$  matrix where element  $(n_1, n_2)$  in this matrix comes from value  $z(n_1, n_2)$ . In eqn. (3.12), we see that, using eqn. (3.8),  $\mathbf{H}$  is an  $N \times (N + M - 1)$  matrix whose  $(n, k)$  element comes from the condition in the curly braces  $\{\}$ . Also note that the first column index into matrix  $\mathbf{H}$  is  $k = -K$  for the purpose of populating this matrix but that typically arrays are indexed either starting from 0 or 1 when programming. This is not a problem as array indexing and the coordinate origin which these indices represent are left to the users' discretion. Now, we assume that  $h$  is a separable filter such that

$$h[m_1, m_2] = h_1[m_1]h_2[m_2]^T \quad (3.13)$$

Then, in the two dimensional case, notice that

$$\begin{aligned} y[n_1, n_2] &= \sum_{k_1=n_1-K_1}^{n_1+K_1} \sum_{k_2=n_2-K_2}^{n_2+K_2} h[n_1 - k_1, n_2 - k_2]x[k_1, k_2] \\ &= \sum_{k_1=n_1-K_1}^{n_1+K_1} \sum_{k_2=n_2-K_2}^{n_2+K_2} h_1[n_1 - k_1]h_2[n_2 - k_2]x[k_1, k_2] \\ &= \sum_{k_1=n_1-K_1}^{n_1+K_1} h_1[n_1 - k_1] \left( \sum_{k_2=n_2-K_2}^{n_2+K_2} x[k_1, k_2]h_2[n_2 - k_2] \right) \end{aligned} \quad (3.14)$$

The parenthesized term in the last line of eqn. (3.14) is equivalent to the 1-D convolution of eqn. (3.10) when working along the  $k_2$  dimension. So eqn. (3.14) just performs a 1-D convolution in the  $k_2$  dimension and then uses this result to perform another 1-D convolution along the  $k_1$  dimension. In matrix/vector form, eqn. (3.14) can be expressed as

$$\mathbf{Y} = \mathbf{H}_1(\mathbf{X}\mathbf{H}_2) \quad (3.15)$$

where

$$\mathbf{H}_1 = \left. \begin{cases} h_1(n_1 - k_1) & -K_1 \leq n_1 - k_1 \leq K_1 \\ 0 & \textit{otherwise} \end{cases} \right\}_{\substack{n_1=0, \dots, N_1-1 \\ k_1=-K_1, \dots, N_1+K_1}} \quad (3.16)$$

and

$$\mathbf{H}_2 = \mathbf{H}_1^T \quad (3.17)$$

Notice that  $\mathbf{H}_2$  is equal to the transpose of  $\mathbf{H}_1$  and that the parenthesized term in eqn. (3.15) represents the parenthesized one in eqn. (3.14). As such,  $\mathbf{H}_1$  has the same form as eqn. (3.12). By applying the vectorizing operator to both sides of eqn. (3.15), we have

$$\begin{aligned} \mathbf{y} &= \textit{vec}(\mathbf{Y}) \\ &= \textit{vec}(\mathbf{H}_1 \mathbf{X} \mathbf{H}_2) \\ &= (\mathbf{H}_2^T \otimes \mathbf{H}_1) \textit{vec}(\mathbf{X}) \\ &= \mathbf{H} \mathbf{x} \end{aligned} \quad (3.18)$$

where  $\textit{vec}(\ )$  is the operator that stacks the columns of a matrix one on top of the other to yield a vector and where “ $\otimes$ ” is the matrix kronecker product [Nand03]. Notice that eqn. (3.18) has been written in the same equation form of eqn. (3.11). However, they are intrinsically different as the convolution matrix  $\mathbf{H}$  in the 2D case is equal to

$$\mathbf{H} = \mathbf{H}_2^T \otimes \mathbf{H}_1 \quad (3.19)$$

### 3.2 Deconvolution by Filtering

Now that we have established our convolution model, we will explain how deconvolution is performed by filtering. These are the standard approaches to deconvolution and are typically the easiest to implement (hence their popularity). However, this ease of implementation usually limits the extent to which these approaches

can be practically used. Nonetheless, their inherent usefulness and point-of-comparison applicability warrants their inclusion. Also, we can gain insights into the deconvolution problem by examining these approaches. We will examine the inverse filter approach as well as the Wiener filter here.

### 3.2.1 The Inverse Filter

In this section, we will find a solution to  $x$  in our finite extent convolution model of eqn. (3.11), by determining the so-called inverse filter. We will explain in more detail how this inverse filter is found in time and also in the frequency domain. Recall that in the previous section 3.1.2, we defined the finite extent convolution of eqn. (3.7) in matrix/vector form as

$$\mathbf{y} = \mathbf{H} \mathbf{x} \quad (3.20)$$

Because  $\mathbf{H}$  in eqn. (3.20) is  $N_1 N_2 \times (N_1 + M_1 - 1)(N_2 + M_2 - 1)$  in size, it is necessarily a rank-deficient matrix. This means there are less rows than columns in eqn. (3.20) which translates to having less equations than unknowns (in  $\mathbf{x}$ ). There is clearly no unique solution for such a case. This is one of the main difficulties in deconvolution. Nonetheless, one of the useful unique solutions to eqn. (3.20) is the minimum-norm one. It is the least squares solution of minimum norm. Thus, the least squares minimum norm solution to eqn. (3.20), i.e.  $\mathbf{x}_{ls}$ , is

$$\mathbf{x}_{ls} = \mathbf{H}^T (\mathbf{H}\mathbf{H}^T)^{-1} \mathbf{y} \quad (3.21)$$

This fact can be easily shown and is given in Appendix A. Notice that multiplying both sides of eqn. (3.21) by  $\mathbf{H}$  results in

$$\mathbf{H} \mathbf{x}_{ls} = \mathbf{H} \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1} \mathbf{y} = \mathbf{y} \quad (3.22)$$

Since  $\mathbf{H} \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1} = \mathbf{I}$  (assuming  $\mathbf{H} \mathbf{H}^T$  is invertible), this implies that the inverse filter must be

$$\mathbf{H}_{inv} = \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1} \quad (3.23)$$

in matrix form. Generally, in all practical FIR filtering applications,  $\mathbf{H}$  is full row rank thus the inverse of  $\mathbf{H} \mathbf{H}^T$  exists. This does not mean that  $\mathbf{x}_{ls}$  is the perfectly recovered image that we seek. It is only the minimum norm (energy) solution that satisfies the convolution relation of eqn. (3.20). In fact, the quality of the solution will depend on the characteristics of the filter  $h$ . This will be addressed a little later. For now, we are interested in determining the inverse FIR filter  $h_{inv}$  to which  $\mathbf{H}_{inv}$  is the matrix counterpart. In this way, we can perform a deconvolution by convolving  $y$  with the inverse filter, i.e.  $\hat{x} = h_{inv} * y = h_{inv} * h * x = x$  if indeed  $h_{inv}$  is the true inverse filter to  $h$  such that  $h_{inv} * h = \delta$  with  $\delta$  representing the unit delta sequence.

In determining the inverse filter, we repeat that  $\mathbf{H} \mathbf{x}$  in eqn. (3.20) is the matrix counterpart to  $h * x$  in the finite extent “valid” convolution model thus  $\mathbf{H}$  is viewed as the “valid” convolution matrix that performs the operation of eqn. (3.14). What then is  $\mathbf{H}^T$ ? As it turns out,  $\mathbf{H}^T$  is the matrix equivalent of a “full” linear correlation filtering. It is essentially the same as the “valid” convolution operation except the filter is not “flipped” and the output resulting from this filtering results in the full linear correlation. Specifically, for a filter  $h$  of support size  $M$  with  $m = -K, \dots, K$  as in eqn. (3.10), the full

linear correlation of  $h$  with a length  $N$  signal  $x$  defined for  $k = 0, \dots, N-1$  would yield an output  $y$  of length  $N + M - 1$  as

$$y[n] = \sum_{k=\max(0, n-K)}^{\min(N-1, n+K)} x[k]h[k-n+K] \quad (3.24)$$

for  $n = -K, \dots, N + K - 1$ . As an example in 1-D, if  $N = 4$  such that  $\mathbf{x} = [x_0, x_1, x_2, x_3]$  and  $K = 1$  such that  $\mathbf{h} = [h_0, h_1, h_2]$ , then  $\mathbf{y} = [x_0h_2, x_0h_1 + x_1h_2, x_0h_0 + x_1h_1 + x_2h_2, x_1h_0 + x_2h_1 + x_3h_2, x_2h_0 + x_3h_1, x_3h_0]$  is the full linear correlation. The significance of this is that  $\mathbf{H}$  multiplied with any vector yields a “valid” or truncated convolution and  $\mathbf{H}^T$  multiplied with any vector yields a full linear correlation. With this in mind,  $\mathbf{H}\mathbf{H}^T \equiv \mathbf{R}$  is a truncated correlation matrix. To see this, consider the autocorrelation sequence of a 3-tap filter  $h = [h_0, h_1, h_2]$  defined for  $n = -1, \dots, 1$  such that

$$\begin{aligned} r &= h \star h \\ &= [h_0, h_1, h_2] \star [h_0, h_1, h_2] \\ &= [h_0h_2, h_0h_1 + h_1h_2, h_0^2 + h_1^2 + h_2^2, h_0h_1 + h_1h_2, h_0h_2] \\ &= [r[-2], r[-1], r[0], r[1], r[2]] \end{aligned} \quad (3.25)$$

where ‘ $\star$ ’ is being used to denote correlation and where eqn. (3.24) was used in obtaining  $r$  by

$$r[n] = \sum_{k=\max(-K, n-K)}^{\min(K, n+K)} h[k]h[k-n+K] \quad (3.26)$$

for  $n = -2K, \dots, 2K$ .

Now, consider the matrix equivalent of  $h \star h$ , i.e.  $\mathbf{H}\mathbf{H}^T$ , so that







Thus, the inverse filter is, approximately (due to truncation)

$$h_{inv} = h \star r_T^{-1} \quad (3.32)$$

In the non approximate case,

$$h_{inv} = h \star r^{-1} \quad (3.33)$$

and notice that, convolving both sides of eqn. (3.33) by  $h$ , we get  $h * h_{inv} = h * (h \star r^{-1})$ . In the case that  $h$  is a symmetric filter, as we will use in this thesis, then autocorrelation is equivalent to convolution and

$$\begin{aligned} h * h_{inv} &= h * (h \star r^{-1}) \\ &= h * (h \star r^{-1}) = (h \star h) \star r^{-1} \\ &= r \star r^{-1} \\ &= \delta \end{aligned} \quad (3.34)$$

where we used the properties in eqn. (3.25), (3.33) and the commutative properties of the linear operators of convolution and correlation.

Now, in the frequency domain the relation between the inverse filter and blurring function  $h$  of eqn. (3.34) is given by the following relation

$$H(\omega)H_{inv}(\omega) = 1 \quad (3.35)$$

as convolution in time is the same as multiplication in the frequency domain. Thus, the inverse filter in the frequency domain is easily solved to be

$$H_{inv}(\omega) = \frac{1}{H(\omega)} \quad (3.36)$$

So, an inverse filter is possible to find in the frequency domain via eqn. (3.36), or in the time domain via eqn. (3.33), as long as the filter is well behaved, i.e. has no frequency nulls or values where  $H(\omega) = 0$ . When dealing with a “badly” behaved filter, in the sense

of deconvolution, only approximate inverse filters can result. We have not shown how to achieve this approximation in the frequency domain as this is not sufficiently relevant to this thesis.

What is more relevant is the effects of noise amplification as a result of inverse filtering. We can view this both in the time and frequency domains. By considering the convolution model of  $y = h * x + \eta$ , we see that inverse filtering results in a deconvolved estimate of  $x$ , called  $\hat{x}$ , that is

$$\begin{aligned}\hat{x} &= h_{inv} * y = h_{inv} * (h * x + \eta) = h_{inv} * h * x + h_{inv} * \eta \\ &= x + h_{inv} * \eta\end{aligned}\tag{3.37}$$

or, in the frequency domain, working with the power spectra of our signals, we have

$$S_{\hat{x}}(\omega) = S_x(\omega) + \frac{S_{\eta}(\omega)}{S_h(\omega)}\tag{3.38}$$

In any case, we can see from eqn. (3.37) or (3.38) that, regardless of the blurring filter's behavior, we cannot avoid noise amplification with inverse filtering. The situation is clearly worse when  $h$  has frequency nulls. In the absence of noise such that  $S_{\eta}(\omega) = 0$ , and as long as the filter is well behaved,  $\hat{x} = x$  so deconvolution with inverse filtering is possible. Even when noise is not absent, as long as the signal-to-noise ratio (SNR) is relatively high, inverse filtering with well behaved filters is a good option. But, when the SNR is not sufficiently high and/or the filter  $h$  is a poorly behaved, alternative approaches to deconvolution are required.

### 3.2.2 The Wiener Filter

The Wiener filter is less sensitive to noise than the inverse filter. This filter is a linear, spatially invariant filter in which the point spread function is the one that minimizes the mean-squared error (MSE) between the ideal and the restored image [Lage00]. This filter minimizes the difference between the ideal and restored images on the average as

$$MSE = E\left[(x[n_1, n_2] - \hat{x}[n_1, n_2])^2\right] \approx \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2 \quad (3.39)$$

where  $\hat{x}[n_1, n_2]$  has the form of eqn. (3.1). The solution of this minimization problem is known as restoration by Wiener filtering. The Wiener filter is typically given in the frequency domain as

$$H_{wiener}(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)}{H^*(\omega_1, \omega_2)H(\omega_1, \omega_2) + \frac{S_N(\omega_1, \omega_2)}{S_X(\omega_1, \omega_2)}} \quad (3.40)$$

where  $H^*(\omega_1, \omega_2)$  is the complex conjugate of the point spread function's frequency response  $H(\omega_1, \omega_2)$  and  $S_X(\omega_1, \omega_2)$  and  $S_N(\omega_1, \omega_2)$  are the power spectra of the ideal image and the noise, respectively. The power spectrum is the average signal power per spatial frequency  $(\omega_1, \omega_2)$  contained in a signal. Using eqn. (3.40), the restored image is then obtained by

$$X_{restored}(\omega_1, \omega_2) = Y(\omega_1, \omega_2)H_{wiener}(\omega_1, \omega_2) \quad (3.41)$$

Notice that the Wiener filter defaults to the inverse filter in the noiseless case, i.e. when  $S_N(\omega_1, \omega_2) = 0$ .

When the degraded image is noisy, the Wiener filter trades-off restoration by inverse filtering with suppression of noise at those frequencies where  $H(\omega_1, \omega_2)$  is close to zero. In general then, the Wiener filter approaches the inverse filter when  $S_N(\omega_1, \omega_2) \ll S_X(\omega_1, \omega_2)$ , and when  $S_N(\omega_1, \omega_2) \gg S_X(\omega_1, \omega_2)$  it acts as a frequency rejection filter, i.e.  $H_{wiener}(\omega_1, \omega_2) \rightarrow 0$ . To find an expression for  $S_N(\omega_1, \omega_2)$ , it is common to assume that the noise is uncorrelated and white. In this case, the power spectrum is determined by the noise variance to be

$$S_N(\omega_1, \omega_2) = \sigma_N^2 \quad \text{for all } (\omega_1, \omega_2) \quad (3.42)$$

In using the Wiener filter, we will observe that for small values of  $\sigma_N^2$ , the restored image will be close to that obtained when inverse filtering. In the case where the noise variance is large, the Wiener filter will over-smooth the restored image. Estimation of  $S_X(\omega_1, \omega_2)$  is problematic since the ideal image  $x$  is clearly never available. Instead, it is typical to approximate  $S_X(\omega_1, \omega_2)$  by an estimate of the power spectrum of the blurred image  $y$  that is available and subsequently compensate for the variance of the noise. That is,

$$S_X(\omega_1, \omega_2) \approx S_Y(\omega_1, \omega_2) - \sigma_N^2 \approx \frac{1}{N_1 N_2} Y^*(\omega_1, \omega_2) Y(\omega_1, \omega_2) - \sigma_N^2 \quad (3.43)$$

Note that  $N_1 \times N_2$  is the size of image  $y$  and that the frequency response is obtained using the discrete time Fourier transform such that

$$Y(\omega_1, \omega_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} y[n_1, n_2] \cdot e^{-j(\omega_1 n_1 + \omega_2 n_2)} \quad (3.44)$$

In chapter 5, we will compare the performance of the Wiener filter and the inverse filter with the method proposed in this thesis.

### 3.3 Implementation Notes

The finite extent convolution model necessarily has more samples in  $x$  that we are solving for than samples in  $y$  that are observed. Specifically, we have said that for a filter of size  $M_1 \times M_2$  and observed image  $y$  of size  $N_1 \times N_2$ , the restored image would be of size  $(N_1 + M_1 - 1) \times (N_2 + M_2 - 1)$ . After any deconvolution restoration, one will notice a larger error around the borders of the restored image compared to the samples in the interior of the restored image. This is due to the relatively smaller number of observed samples in  $y$  that any sample of  $x$  near a border can affect, hence there is less reliability in these border pixels' estimation. Because of this, the restored image is most reliably determined to be those interior  $N_1 \times N_2$  image samples in the  $(N_1 + M_1 - 1) \times (N_2 + M_2 - 1)$  samples that were solved for. As such, evaluation of quality of restoration should only make use of those “reliably restored” samples in  $\hat{x}$  against their original samples in  $x$ .

Another important implementation note concerns the Wiener filter. That is, it is important to keep the finite extent convolution model in mind when Wiener filtering. This is the reason that “ringing” occurs near the borders of the restored image. That is, keep in mind that the Wiener filter is approximately the inverse filter of the original blurring filter. As such, it has a usually longer region of support than  $h$  and when “convolved” with the degraded image, results in the oscillations usually evident near the borders.

In any case, we can see for eqn. (3.37) or (3.38) that, regardless of the blurring filter's behavior, we cannot avoid noise amplification with inverse filtering. The situation is clearly worse when  $\mathbf{H}$  has frequency nulls. In the absence of noise  $S_N(\omega_1, \omega_2) = 0$  and as long as the filter is well behaved,  $\hat{x} = x$  so deconvolution with inverse filtering is possible. Even when noise is not absent, as long as the signal-to-noise ratio (SNR) is relatively high, inverse filtering with a well behaved filter is a good option. But, when the SNR is not sufficiently high and/or the filter  $h$  is a poorly behaved, alternative approaches to deconvolution are required.

## Chapter 4

### Noise and Range Constrained Deconvolution

In the two standard deconvolution approaches presented in chapter 3, the effects of noise largely influenced the resulting solution. In the case of inverse filtering, noise amplification becomes increasingly unacceptable as the SNR decreases. In the case of Wiener filtering, noise amplification is controlled but at the expense of unacceptable smoothing on the restored image. In effect, noise is the main limiting factor to successful deconvolution with the point spread functions' nulling characteristics being the second (and usually not as serious) limiting factor. There is no way to get rid of noise, just different ways to mitigating its effects. The key is to restore the image without having the noise be modified in an unacceptable manner. In this chapter, we present two techniques that suppress the amplification of noise and though not removing it, aids in maintaining the noise level while the image is restored.

The techniques we present involve constraining the space of pixel values acceptable in the solution as well as constraining the moments of the error signal (difference between the degraded image and the re-blurred estimate of  $x$ ) to coincide statistically with the characteristics of the noise. The original motivation for the range constraint was that if the lens effects or source of blurring had not been present, we could have captured a sharp image at the pixel precision allowable by the camera. That is, if we were working with an 8 bit per pixel device, then our sharp image would have also been represented with this precision. As it turns out, this constraint indirectly helps control the amplification of noise. The motivation for the noise moment constraint was to not

amplify noise in the solution while the deblurring was taking place. It seemed reasonable to enforce this via the use of a random variable's moments rather than through its probability density function. We now detail the constraints used and discuss how we perform the constrained deconvolution optimally in an unconstrained manner.

#### 4.1 Range Constraint

In this thesis, a non-linear variable transformation is used to simplify the solution of the signal  $x$  while enforcing a range-limiting constraint. In our implementation, we have elected to work with normalized pixel values such that  $0 \leq x \leq 1$ . Although this range interval is arbitrary, it is mathematically convenient. In our development, we will express the original image  $x$  as the nonlinear mapping  $f$  of an auxiliary variable  $\chi$  as

$$x = f(\chi) \tag{4.1}$$

where  $f$  is equal to

$$f(\chi) = \frac{1}{2} \tanh(\chi) + \frac{1}{2} \tag{4.2}$$

As such, our solution will be in terms of the auxiliary variable  $\chi$ , not  $x$ . The usefulness of this auxiliary parameterization is that  $\chi$  can take a value  $-\infty < \chi < \infty$  and so there are no constraints in its value. And, since the mapping  $f(\chi)$  does yield the desired range for  $x$ , we can indirectly constrain the solution without directly enforcing this constraint in the optimization's procedure – thus simplifying the deconvolution approach in general.

#### 4.2 Noise Moment Constraints

Having prior knowledge of the statistics of the additive noise corrupting the blurred image is important because it can improve the performance of the deconvolution

process. It also serves as an objective means of establishing a stopping criteria to iterative deconvoluting approaches – in contrast to typical subjective approaches. This will be addressed later. These moment constraints are a new means of allowing for the constraining of the deconvolution solution. In this approach, we desire that the error vector exhibit samples consistent with the noise distribution  $\eta$ . We accomplish this by ensuring that one or more moment estimates from the error signal match the true moments of  $\eta$ .

Two types of moment constraints on the additive noise  $\eta$  will be mentioned in this section: those using standard moments and those involving probability weighted moments (PWM). Standard moments use unbiased statistical estimators on the samples from the convolution error signal. The statistical standard moments that we are used to seeing are given by

$$m_p = E[\eta^p] = \int_{-\infty}^{\infty} \eta^p f_{\eta}(\eta) d\eta \quad (4.3)$$

where  $E[\cdot]$  is the expectation operator [Papo02],  $m_p$  is the  $p$ th moment of  $\eta$ , and  $f_{\eta}(\eta)$  is the probability density function of random variable  $\eta$ . The unbiased sample estimator of these moments is given by

$$\hat{m}_p = \frac{1}{N} \sum_{n=0}^{N-1} e^p[n] \quad (4.4)$$

with  $\hat{m}_p$  representing the estimate of  $m_p$  where the hat ‘^’ indicates “estimate” and where the error samples  $e[n]$  in our convolution are interpreted as samples of the noise distribution  $\eta$ .

Because the use of standard moments in the optimization is computationally involved, probability weighted moments (PWM) provide an alternative moment definition that we can use. As will be described, they simplify the mathematical complexity related with the moment estimation computations. Their attractiveness lies in the linear transformation used to obtain the estimate. This is in contrast to the nonlinear, i.e. exponentiating, operation required in obtaining  $\hat{m}_p$  in eqn. (4.4). The PWM moment estimates  $\hat{m}_p$  are found with the following linear transformation on the error samples:

$$\hat{m}_p = \sum_{n=1}^N b_p[n]e_o[n] = \mathbf{b}_p^T \mathbf{e}_o \quad (4.5)$$

where  $\mathbf{b}_p = [b_p[1], \dots, b_p[N]]^T$  is the probability weighted  $p^{\text{th}}$  moment estimator vector, and  $\mathbf{e}_o = [e_o[1], \dots, e_o[N]]^T$  is a vector with the samples of the error signal *ordered* in ascending order. The ordering comes from the rank-order statistics that the PWMs are based upon. These moments are now further detailed.

### 4.3 The Probability Weighted Moments

In this section, the PW moments are defined and described. The PW moments of a uniform and a Gaussian distribution will then be derived and detailed. The  $r^{\text{th}}$  PW moment<sup>1</sup>,  $\beta_r$ , of an arbitrary distribution is defined as

$$\beta_r = E[xF_x^r(x)] = \int_{-\infty}^{\infty} xF_x^r(x)f_x(x)dx \quad (4.6)$$

---

<sup>1</sup> There are other probability weighted moment definitions but the one we present is convenient for our deconvolution approach. See [Gree79, Land79] for the general PW moment definition.

where, for RV  $X$ ,  $f_x(x)$  is its probability density function (PDF) and  $F_x(x)$  is its cumulative distribution function (CDF). This is in contrast to the typical  $r^{\text{th}}$  moment of RV  $X$  that was defined as

$$m_r = E[x^r] = \int_{-\infty}^{\infty} x^r f_x(x) dx \quad (4.7)$$

in eqn. (4.3) and is reproduced here for convenience.

#### 4.3.1 Uniformly Distributed PW Moments

The probability weighted moments of a uniform distribution are found by substituting the CDF and the PDF of the uniform distribution into eqn. (4.6). We note that the CDF  $F_x(x)$  of a uniformly distributed RV  $X$  is

$$F_x(x) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a < x < b \\ 1 & b \leq x \end{cases} \quad (4.8)$$

and its PDF  $f_x(x)$  is

$$f_x(x) = \begin{cases} 1/(b-a) & a \leq x < b \\ 0 & \text{otherwise} \end{cases} \quad (4.9)$$

Using equations (4.8) and (4.9) in (4.6), we obtain

$$\beta_r = \int_a^b x [F_x(x)]^r f_x(x) dx = \int_a^b x \left( \frac{x-a}{b-a} \right)^r \frac{1}{b-a} dx \quad (4.10)$$

Since  $a$  and  $b$  are constant, they can be moved out of the integral yielding

$$\beta_r = \frac{1}{(b-a)^{r+1}} \int_a^b x(x-a)^r dx \quad (4.11)$$

then, letting  $y = x - a$  so that  $dy = dx$ , we have

$$\begin{aligned}
\beta_r &= \frac{1}{(b-a)^{r+1}} \int_a^b (y+a)y^r dy \\
&= \frac{1}{(b-a)^{r+1}} \int_0^{b-a} (y^{r+1} + ay^r) dy \\
&= \frac{1}{(b-a)^{r+1}} \left[ \frac{y^{r+2}}{r+2} + a \cdot \frac{y^{r+1}}{r+1} \Big|_0^{b-a} \right] \\
&= \frac{1}{(b-a)^{r+1}} \left[ \frac{(b-a)^{r+2}}{r+2} + a \cdot \frac{(b-a)^{r+1}}{r+1} \right] \\
&= \frac{b-a}{r+2} + \frac{a}{r+1} = \frac{(r+1)(b-a) + a(r+2)}{(r+2)(r+1)} \\
&= \frac{rb + b - ar - a + ar + 2a}{(r+2)(r+1)} \\
&= \frac{rb + b + a}{(r+2)(r+1)}
\end{aligned} \tag{4.12}$$

As an example, the first ten PW moments of a uniform distribution over the unit interval [0,1] are shown in Table 4-1.

**Table 4-1 First ten PW moments of a uniform distribution on the unit interval**

$r$	$\beta_r$
1	0.5000
2	0.3333
3	0.2500
4	0.2000
5	0.1667
6	0.1429
7	0.1250
8	0.1111
9	0.1000
10	0.0909

### 4.3.2 Gaussian Distributed PW Moments

For a Gaussian distribution, its CDF, of which there is no closed form, is

$$F_x(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-(u-\mu)^2/2\sigma^2} du . \tag{4.13}$$

Its PDF is given by

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} \quad (4.14)$$

and note this distribution is parameterized by its mean  $\mu$  and standard deviation  $\sigma$ .

Substituting eqn. (4.13) and (4.14) into eqn. (4.6) yields

$$\begin{aligned} \beta_r &= \int_{-\infty}^{\infty} x F_x^r(x) f_x(x) dx \\ &= \int_{-\infty}^{\infty} x \left[ \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-(u-\mu)^2/2\sigma^2} du \right]^r \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2} dx \\ &= \frac{1}{\sqrt{(2\pi)^{r+1} \sigma^2}} \int_{-\infty}^{\infty} x \left[ \int_{-\infty}^x e^{-(u-\mu)^2/2\sigma^2} du \right]^r e^{-(x-\mu)^2/2\sigma^2} dx \end{aligned} \quad (4.15)$$

It is observed that there is no closed form solution for this integral. It can however be approximated numerically. Table 4-2 shows the first ten PW moments of the standard normal Gaussian distribution. i.e.  $X \sim N(0,1)$ .

**Table 4-2 First ten PW moment of a standard normal Gaussian distributed function**

$r$	$\beta_r$
1	0
2	0.2821
3	0.2821
4	0.2573
5	0.2326
6	0.2112
7	0.1982
8	0.1780
9	0.1650
10	0.1539

Note that when dealing with a general Gaussian model, i.e.  $X \sim N(\mu, \sigma)$ , the PW moments are obtained following the procedure developed by Hoskings and described in Appendix B.

#### 4.4 Estimating Probability Weighted Moments

The unbiased estimate of the  $r^{\text{th}}$  PWM, computed from  $n$  data values  $x_1, x_2, \dots, x_N$  such that  $x_n \leq x_{n+1}$ , i.e., they are sorted in increasing order, is given by

$$\hat{m}_r = \begin{cases} \frac{1}{N} \sum_{j=1}^N x_j & r = 0 \\ \frac{1}{N} \sum_{j=r+1}^N \frac{(j-1)(j-2)\dots(j-r)}{(N-1)(N-2)\dots(N-r)} x_j & r = 1, \dots, N \end{cases} \quad (4.16)$$

This is in contrast to those moment estimates found in eqn. (4.4) of the standard moments. We can see that  $\hat{m}_r$  in eqn. (4.16) results from a linear combination of data samples, hence the inner product of eqn. (4.5).

From the moment estimates in eqn. (4.16) we can conveniently determine a moment estimator matrix that can estimate the  $N$  PW moments of a sample vector  $\mathbf{x}$  of length  $N$ . This moment estimation is written in matrix form as

$$\hat{\mathbf{m}} = \mathbf{B} \mathbf{x}_0 \quad (4.17)$$

where  $\mathbf{x}_0 = [x_1, x_2, \dots, x_N]^T$  is the vector of sorted data samples. The moment estimator matrix is an upper diagonal matrix with all elements below the diagonal being zero. We establish the elements in the moment estimator matrix as in Algorithm 1. If we assume matrix  $\mathbf{B}$ 's indices  $i, j$  start from zero, then an algorithm for computing the moment estimator matrix  $\mathbf{B}$  for a length  $N$  sample vector is

Algorithm 1:

```
for i = 0,...,N-1
  for j = i,...,N-1
    if i = 0
       $\mathbf{B}[i, j] = \frac{1}{N}$ 
    else
       $\mathbf{B}[i, j] = \frac{(j)(j-1)\cdots(j-i+1)}{(N-1)\cdots(N-i)} \cdot \frac{1}{N}$ 
    end
  end
end
end
```

where  $N$  is equal to the length of our data vector  $\mathbf{x}$ . As an example, the first 3 rows and columns of the moment estimator matrix  $\mathbf{B}$  for an  $N=5$  length data vector is

$$\mathbf{B}[0:2,0:2] = \begin{bmatrix} 0.2000 & 0.2000 & 0.2000 \\ 0 & 0.0500 & 0.1000 \\ 0 & 0 & 0.0333 \end{bmatrix}$$

#### 4.5 Performing the Deconvolution

The deconvolution process will be setup as a constrained optimization. We do so by establishing a cost that minimizes the energy of the error signal while enforcing constraints such that this error signal's samples approximately correspond to a desired distribution. This optimization process proceeds by first establishing a Lagrangian function. This Lagrangian function is composed of a cost function and established constraints and is parameterized by the auxiliary image  $\chi$  (described in eqn. (4.1)) and the Lagrange multipliers  $\lambda$ . Its form is

$$\begin{aligned}
L(\chi, \lambda) &= J(\chi) + \lambda_1 C_1(\chi) + \lambda_2 C_2(\chi) + \dots + \lambda_M C_M(\chi) \\
&= \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^2[n_1, n_2] + \sum_{p=0}^{M-1} \lambda_p \left( \hat{m}[p] - m[p] \right)
\end{aligned} \tag{4.18}$$

where

$$J(\chi) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^2[n_1, n_2] \tag{4.19}$$

is the mean squared error of our estimation and

$$C_p(\chi) = \hat{m}[p] - m[p] \tag{4.20}$$

is the  $p^{\text{th}}$  constraint equation such that the error signal's samples (which will only correspond to noise samples at the true solution) agrees, via moments, with the noise model we have assumed.

Notice that the  $\chi$  shows up in the error signal since the error signal is equal to  $e = y - \hat{y} = y - h * x = y - h * f(\chi)$ , where  $y$  is the true image and  $\hat{y}$  is the estimated signal described in eqn. (3.14). And, because the error signal is a part of eqn. (4.19) and also (4.20) by way of the moment estimate, the notation  $J(\chi)$  and  $C_p(\chi)$  was used. The first term in eqn. (4.18) is the cost to minimize and any constraints are included by way of Lagrange multipliers  $\lambda_p$ .

An optimal minimum, solution is found when the vector pair  $\{\lambda, \chi\}$  minimizes  $L$  where  $\lambda = [\lambda_1, \dots, \lambda_M]$  and

$$\chi = \text{vec} \left( \left\{ \chi[m_1, m_2] \right\}_{\substack{m_1=-K_1, \dots, N+K_1-1 \\ m_2=-K_2, \dots, N+K_2-1}} \right). \tag{4.21}$$

At least for a local minimum, this requires that the gradient of  $L$  with respect to both  $\chi$  and  $\lambda$  be zero, i.e.  $\nabla_{\chi} L = \mathbf{0}$  and  $\nabla_{\lambda} L = \mathbf{0}$ . In general, minimizing  $L$  is increasingly

complex as the number of constraints  $M$  increases. In actuality, if all constraints can not be satisfied, a solution that trades off constraints is reached. Note also that the choice of moments used will affect the final form of eqn. (4.18) as the moment estimator  $\hat{m}_p$  will change. We can use either eqn. (4.4) or (4.5) in this regard. Finally, note that the Lagrangian of eqn. (4.18) can be expressed in matrix/vector form as

$$L(\chi, \lambda) = \mathbf{e}^T \mathbf{e} + \lambda^T (\hat{\mathbf{m}} - \mathbf{m}) \quad (4.22)$$

where  $\mathbf{e} = \text{vec}(e[n_1, n_2])$ ,  $\hat{\mathbf{m}} = [\hat{m}[1], \dots, \hat{m}[M]]$  and  $\mathbf{m} = [m[1], \dots, m[M]]$ .

#### 4.6 Optimization Using Standard Moment Constraints

Using eqn. (4.18) with standard moments as constraints, we can substitute the 2-D form of eqn. (4.4) into eqn. (4.18) to get

$$L(\chi, \lambda) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^2[n_1, n_2] + \sum_{p=0}^{M-1} \lambda_p \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right)^2 \quad (4.23)$$

We square the constraint in eqn. (4.23) because the Lagrangian function must be twice continuously differentiable. This is a condition for any local minima-Lagrange multiplier pair to be a point of attraction to gradient descent-type optimization [Bert95]. Also, by squaring the constraints, we can have better control over the oscillatory behavior in each iteration. This type of cost and constraints has been used previously in [Cand04].

To find an optimal minimum solution, we find the vector pair  $\{\lambda, \chi\}$  that minimizes  $L$  (at least locally). That is, we must satisfy  $\nabla_{\lambda} L = \mathbf{0}$  and  $\nabla_{\chi} L = \mathbf{0}$ . The gradient of  $L$  with respect to  $\chi$  is found by determining

$$\frac{\partial L(\chi, \lambda)}{\partial \chi[m_1, m_2]} = \frac{\partial L(\chi, \lambda)}{\partial x[p_1, p_2]} \cdot \frac{\partial x[p_1, p_2]}{\partial \chi[m_1, m_2]} \quad (4.24)$$

where we see that

$$\begin{aligned} \frac{\partial L(\chi, \lambda)}{\partial x[p_1, p_2]} &= \underbrace{\frac{\partial}{\partial x[p_1, p_2]} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^2[n_1, n_2]}_{(1)} \\ &+ \underbrace{\frac{\partial}{\partial x[p_1, p_2]} \left( \sum_{p=0}^{M-1} \lambda_p \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right)^2 \right)}_{(2)} \end{aligned} \quad (4.25)$$

To simplify our analysis, we divide eqn. (4.25) into two terms, (1) and (2). Then the derivative of term (1) with respect to  $x$  is equal to

$$\begin{aligned} \frac{\partial}{\partial x[p_1, p_2]} \left( \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^2[n_1, n_2] \right) &= \frac{\partial}{\partial x[p_1, p_2]} \left( \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y[n_1, n_2] - \hat{y}[n_1, n_2])^2 \right) \\ &= 2 \cdot \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y[n_1, n_2] - \hat{y}[n_1, n_2]) \cdot \frac{\partial}{\partial x[p_1, p_2]} (y[n_1, n_2] - \hat{y}[n_1, n_2]) \quad (4.26) \\ &= -2 \cdot \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y[n_1, n_2] - \hat{y}[n_1, n_2]) \cdot \frac{\partial}{\partial x[p_1, p_2]} (\hat{y}[n_1, n_2]) \end{aligned}$$

where the derivative of the estimate of  $y$  is equal to

$$\begin{aligned} \frac{\partial}{\partial x[p_1, p_2]} (\hat{y}[n_1, n_2]) &= \frac{\partial}{\partial x[p_1, p_2]} \left( \sum_{k_1=n_1-K_1}^{n_1+K_1} \sum_{k_2=n_2-K_2}^{n_2+K_2} h[n_1 - k_1, n_2 - k_2] x[k_1, k_2] \right) \\ &= h[n_1 - p_1, n_2 - p_2] \end{aligned} \quad (4.27)$$

Then, by substituting eqn. (4.27) into eqn. (4.26), term (1) yields

$$\frac{\partial}{\partial x[p_1, p_2]} \left( \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^2[n_1, n_2] \right) = -2 \cdot \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (y[n_1, n_2] - \hat{y}[n_1, n_2]) \cdot h[n_1 - p_1, n_2 - p_2] \quad (4.28)$$

The derivative of term (2) with respect to  $x$  is equal to

$$\begin{aligned}
& \frac{\partial}{\partial x[p_1, p_2]} \sum_{p=0}^{M-1} \lambda_p \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right)^2 \\
&= \sum_{p=0}^{M-1} \lambda_p \cdot 2 \cdot \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right) \cdot \\
&\quad \left( \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \frac{\partial}{\partial x[p_1, p_2]} (y[n_1, n_2] - \hat{y}[n_1, n_2])^p \right) \\
&= \sum_{p=0}^{M-1} \lambda_p \cdot 2 \cdot \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right) \cdot \\
&\quad \left( \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} p \cdot (y[n_1, n_2] - \hat{y}[n_1, n_2])^{p-1} \frac{\partial}{\partial x[p_1, p_2]} \hat{y}[n_1, n_2] \right) \\
&= \sum_{p=0}^{M-1} \lambda_p \cdot 2 \cdot \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right) \cdot \\
&\quad \left( \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} p \cdot e^{(p-1)}[n_1, n_2] \cdot h[n_1 - p_1, n_2 - p_2] \right) \\
&= 2 \sum_{p=0}^{M-1} p \cdot \lambda_p \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right) \left( \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^{(p-1)}[n_1, n_2] h[n_1 - p_1, n_2 - p_2] \right)
\end{aligned} \tag{4.29}$$

Thus, the gradient of  $L$  with respect to  $x$  is found by substituting (4.28) and (4.29) into (4.25). This yields

$$\begin{aligned}
\frac{\partial L(x, \lambda)}{\partial x[p_1, p_2]} &= -2 \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e[n_1, n_2] h[n_1 - p_1, n_2 - p_2] \\
&\quad + 2 \sum_{p=0}^{M-1} p \cdot \lambda_p \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right) \left( \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^{(p-1)}[n_1, n_2] h[n_1 - p_1, n_2 - p_2] \right)
\end{aligned} \tag{4.30}$$

This is the gradient needed for a moment-only constrained solution, i.e. no range constraints have been considered. If a range constrained solution is sought, then notice that

$$\begin{aligned}
\frac{\partial L(\chi, \lambda)}{\partial \chi[m_1, m_2]} &= \frac{\partial L(\chi, \lambda)}{\partial x[p_1, p_2]} \cdot \frac{\partial x[p_1, p_2]}{\partial \chi[m_1, m_2]} \\
&= -2 \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e[n_1, n_2] h[n_1 - m_1, n_2 - m_2] \\
&\quad + 2 \sum_{p=0}^{M-1} p \cdot \lambda_p \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right) \\
&\quad \left( \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^{(p-1)}[n_1, n_2] h[n_1 - m_1, n_2 - m_2] \right) \cdot \frac{\partial f(\chi[p_1, p_2])}{\partial \chi[m_1, m_2]}
\end{aligned} \tag{4.31}$$

where

$$\frac{\partial f(\chi[p_1, p_2])}{\partial \chi[m_1, m_2]} = \frac{\partial x[p_1, p_2]}{\partial \chi[m_1, m_2]} = 2 \cdot f(\chi[m_1, m_2]) \cdot [1 - f(\chi[m_1, m_2])] \tag{4.32}$$

because each pixel  $x[p_1, p_2]$  is only related to  $\chi[p_1, p_2]$ . The nonlinear mapping was given previously in eqn. (4.2) and note that the derivative form of eqn. (4.32) is derived in Appendix C. Finally, the gradient of  $L$  with respect to  $\lambda_p$  is

$$\frac{\partial L(\chi, \lambda)}{\partial \lambda_p} = \left( \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} e^p[n_1, n_2] \right] - m[p] \right)^2 \tag{4.33}$$

Making use of the gradients in eqns. (4.31) and (4.33), we solve for  $\chi$  using the conjugate gradient approach as described in [Cand04].

#### 4.7 Optimization Using Probability Weighted Moment Constraints

In this section, it is useful to address the optimization with PW moments in matrix/vector notation as both the one and the two dimensional cases can be described in this single notation. We will use the notation  $\hat{m}_p$  or  $\hat{m}[p]$  interchangeably to represent

the estimates of the  $p^{\text{th}}$  probability weighted moment. These estimates, previously described in eqns. (4.16) and (4.17), are written

$$\hat{m}_p = \hat{m}[p] = \sum_{j=0}^{N_1 N_2 - 1} \mathbf{B}[p, j] \mathbf{e}_o[j] \quad (4.34)$$

for  $p = 0, \dots, M - 1$  and where  $N_1 N_2$  is the total number of samples in our image  $y$ ,  $\mathbf{B}$  is the matrix of Algorithm 1 and  $\mathbf{e}_o$  is the sequence of samples of the error signal  $\mathbf{e}$  ordered in ascending order, i.e.,

$$\mathbf{e}_o[j] = \mathbf{e}[\text{indx}[j]] \quad (4.35)$$

where  $\text{indx}[j]$ ;  $j = 0, \dots, N_1 N_2 - 1$  represents the mapping (or re-ordering) of indices  $j$  so that the samples in  $e$  can be sorted in ascending order. Eqn. (4.35) is also expressed in matrix/vector form as

$$\mathbf{e}_o = \mathbf{P} \mathbf{e} \quad (4.36)$$

where  $\mathbf{P}$  represents a permutation matrix. This permutation matrix performs the re-ordering mapping of  $\text{indx}[j]$  that sorts the samples in the error vector signal. For example, if  $\mathbf{e} = [3, 2, 1, 4]^T$  is a vector of error samples and  $\mathbf{e}_o = [1, 2, 3, 4]^T$  is its sorted counterpart, then using eqn. (4.36), we would have

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

in order to achieve our ordering.

In following our matrix/vector convention, recall that  $\hat{\mathbf{m}} = [\hat{m}_1, \dots, \hat{m}_M]$ ,  $\mathbf{e} = \text{vec}(e[n_1, n_2])$ ,  $\mathbf{y} = \text{vec}(y[n_1, n_2])$  and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}[0,0] & \mathbf{B}[0,1] & \cdots & \mathbf{B}[0, N_1 N_2 - 1] \\ 0 & \mathbf{B}[1,1] & \cdots & \mathbf{B}[1, N_1 N_2 - 1] \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{B}[M-1, N_1 N_2 - 1] \end{bmatrix}.$$

With these definitions, the Lagrangian function of eqn. (4.18) is expressed in matrix form as follows:

$$\begin{aligned} L(\boldsymbol{\chi}, \boldsymbol{\lambda}) &= J(\boldsymbol{\chi}) + \boldsymbol{\lambda}^T C(\boldsymbol{\chi}) \\ &= \mathbf{e}^T \mathbf{e} + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{P}\mathbf{e} - \mathbf{m}) \\ &= \mathbf{e}_o^T \mathbf{e}_o + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{e}_o - \mathbf{m}) \end{aligned} \quad (4.37)$$

where eqn. (4.17) and (4.36) were inserted into eqn. (4.18) to yield the last term on the right hand side. Notice that the sum of squared elements in a vector yields the same answer whether those elements were sorted or not, hence  $\mathbf{e}^T \mathbf{e} = \mathbf{e}_o^T \mathbf{e}_o$ . And, since the expression for the error vector using eqn. (3.9) yields

$$\begin{aligned} \mathbf{e} &= \mathbf{y} - \hat{\mathbf{y}} \\ &= \mathbf{y} - \mathbf{H} \cdot \mathbf{x} \end{aligned} \quad (4.38)$$

we can substitute this in eqn. (4.37), noting that  $\mathbf{e}_o = \mathbf{P}\mathbf{e}$ , to get

$$\begin{aligned} L(\boldsymbol{\chi}, \boldsymbol{\lambda}) &= [\mathbf{P}(\mathbf{y} - \mathbf{H}\mathbf{x})]^T [\mathbf{P}(\mathbf{y} - \mathbf{H}\mathbf{x})] + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{P}(\mathbf{y} - \mathbf{H}\mathbf{x}) - \mathbf{m}) \\ &= \mathbf{y}^T \mathbf{P}^T \mathbf{P} \mathbf{y} - 2\mathbf{y}^T \mathbf{P}^T \mathbf{P} \mathbf{H} \mathbf{x} + \mathbf{x}^T \mathbf{H}^T \mathbf{P}^T \mathbf{P} \mathbf{H} \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{P}\mathbf{y} - \mathbf{m} - \mathbf{B}\mathbf{P}\mathbf{H}\mathbf{x}) \\ &= \mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \mathbf{H} \mathbf{x} + \mathbf{x}^T \mathbf{H}^T \mathbf{H} \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{B}\mathbf{P}\mathbf{y} - \mathbf{m} - \mathbf{B}\mathbf{P}\mathbf{H}\mathbf{x}) \end{aligned} \quad (4.39)$$

The last equality resulted from  $\mathbf{P}^T \mathbf{P} = \mathbf{I}$ . Please refer to proof in Appendix D.

Having established the form of our Lagrangian function and noting that this is in quadratic form, a globally optimal solution can be found. This is obtained by solving

$$\nabla_{\boldsymbol{\chi}} L = \mathbf{0} \quad (4.40)$$

and

$$\nabla_{\lambda} L = \mathbf{0} \quad (4.41)$$

From eqn. (4.39) and (4.40), we require

$$\nabla_{\mathbf{x}} L = (-2\mathbf{y}^T + 2\mathbf{x}^T \mathbf{H}^T + \lambda^T \mathbf{B}\mathbf{P})\mathbf{H} = \mathbf{0} \quad (4.42)$$

and

$$\nabla_{\lambda} L = \mathbf{B}\mathbf{P}\mathbf{e} - \mathbf{m} = \mathbf{0}. \quad (4.43)$$

Now, from eqn. (4.42), we solve for the Lagrange multipliers and find that

$$\lambda^T = -2(\mathbf{y} - \mathbf{H}\mathbf{x})^T \mathbf{P}^T \mathbf{B}^T (\mathbf{B}\mathbf{B}^T)^{-1} \quad (4.44)$$

Then, since  $\mathbf{B}\mathbf{P}\mathbf{e} = \mathbf{m}$ , which is also equal to

$$\mathbf{B}\mathbf{P}(\mathbf{y} - \mathbf{H}\mathbf{x}) = \mathbf{m} \quad (4.45)$$

we can substitute eqn. (4.45) into eqn. (4.44) and we see that

$$\lambda^T = -2\mathbf{m}^T (\mathbf{B}\mathbf{B}^T)^{-1} \quad (4.46)$$

Notice that the Lagrange multipliers  $\lambda$  only depend on prior knowledge of the noise distribution since  $\lambda$  only depends on the true moment vector  $\mathbf{m}$ . Now, having solved for  $\lambda$ , we only need to solve for  $\mathbf{x}$  in eqn. (4.42). In the case where we choose to include range constraints,  $\mathbf{x}$  is related to  $\chi$  by a nonlinear mapping (as explained in eqns. (4.1) and (4.2)) and a closed form solution is not available. Instead,  $\chi$  is found by using an iterative gradient method such as the steepest descent method (where we will later explain the reason for choosing this particular method). In the method of steepest descent, also called the gradient descent method, we start at a point  $\chi_0$  which is an initial guess of our solution. We then update  $\chi$  proportionally to the gradient of our last guess  $\chi_k$  as

$$\chi_{k+1} = \chi_k - \alpha \nabla_{\chi} L(\chi_k) \quad (4.47)$$

A good choice of initial guess uses the image that we are given. That is, given an image  $y$  and the nonlinear mapping of eqn. (4.2), we let

$$\chi_0 = f^{-1}(y) = \tanh^{-1}\left(2\left(y - \frac{1}{2}\right)\right) \quad (4.48)$$

Note that  $y$  must have samples values in between (0,1) in order to prevent complex and/or infinite values when evaluating  $\tanh^{-1}$ . It also important to note that  $y$  is an  $N_1 \times N_2$  sample sequence but  $\chi$  will be  $(N_1 + M_1 - 1) \times (N_2 + M_2 - 1)$  in size. Since  $\chi$  has more samples than  $y$ , we use a mirrored extension of  $y$  in all direction such that  $(N_1 + M_1 - 1) \times (N_2 + M_2 - 1)$  samples results. This is used in eqn. (4.48) to generate the initial values of  $\chi_0$ .

In the implementation, we note that  $\alpha$  in eqn. (4.47) is a small constant and  $\nabla_{\chi} L(\chi_k)$  is equal to

$$\begin{aligned} \nabla_{\chi} L(\chi_k) &= \nabla_{\mathbf{e}} L \cdot \nabla_{\mathbf{x}} \mathbf{e} \cdot \nabla_{\chi} \mathbf{x} \\ &= [2\mathbf{e}^T + \lambda^T \mathbf{B}\mathbf{P}] \cdot [-\mathbf{H}] \cdot \text{diag}(f'(\chi_k)) \end{aligned} \quad (4.49)$$

where  $\text{diag}(\mathbf{z})$  represents a diagonal matrix whose elements are given by the vector  $\mathbf{z}$  and

where we have used the facts that  $\nabla_{\mathbf{e}} L = [2\mathbf{e}^T + \lambda^T \mathbf{B}\mathbf{P}]$ ,  $\nabla_{\mathbf{x}} \mathbf{x} = \text{diag}(f'(\chi_k))$  and

$\nabla_{\mathbf{x}} \mathbf{e} = -\mathbf{H}$ . As such, we obtain

$$\begin{aligned} \nabla_{\chi} L(\chi_k) &= [2\mathbf{e}^T - 2\mathbf{m}^T (\mathbf{B}\mathbf{B}^T)^{-1} \mathbf{B}\mathbf{P}] \cdot [-\mathbf{H}] f'(\chi_k) \\ &= 2 \left[ (\mathbf{m}^T (\mathbf{B}\mathbf{B}^T)^{-1} \mathbf{B}\mathbf{P} - \mathbf{e}^T) \mathbf{H} \right] \circ f'(\chi_k) \\ &= 2 f'(\chi_k) \circ \left[ (\mathbf{m}^T (\mathbf{B}\mathbf{B}^T)^{-1} \mathbf{B}\mathbf{P} - \mathbf{e}^T) \mathbf{H} \right] \end{aligned} \quad (4.50)$$

where “ $\circ$ ” represents the Schur product or element by element matrix multiplication and  $f'(\chi_k)$  is the derivative vector of our range mapping  $f(\chi)$  of eqn. (4.2). The derivative of this mapping function is given by

$$f'(\chi) = \frac{1}{2}(1 - \tanh^2(\chi)) \quad (4.51)$$

This equation is equivalent to (see Appendix C for derivation)

$$f'(\chi) = 2 \cdot f(\chi) \cdot (1 - f(\chi)) \quad (4.52)$$

Thus, with the iterating procedure of eqn. (4.47), the initialization of eqn. (4.48), the gradient of eqn. (4.50) and the derivative range mapping of eqn. (4.52), the deconvolution operator based on PWMs can be performed.

We now note some important points:

1. Because of the nature of PW moments, we require any data vector to be sorted prior to estimating the moments. In our implementation we have a sorting operation that performs the equivalent role of the permutation matrix  $\mathbf{P}$ .
2. The Lagrangian of eqn. (4.39) has a constraint of  $\mathbf{BPe} - \mathbf{m}$ . This means, at the solution for  $\mathbf{x}$ , we take our error vector  $\mathbf{e} = \mathbf{y} - \mathbf{Hx}$  and sort out the samples (via the  $\mathbf{Pe}$  operation). The problem is that we would need to know the solution in order to obtain the correct sorting order for  $\mathbf{P}$  and this is clearly not possible. And, to obtain the correct noise estimates, one needs to sort the error samples (this operation cannot be ignored).
3. Since we need to sort samples for our PW moment estimates, it seems that a reasonable way of arriving at the solution for  $\mathbf{x}$  is to slowly perform updates to  $\mathbf{x}$  (or similarly  $\chi$ ) while making sure the PW moment constraints on the error signal

are being satisfied. Thus, along the way, we will be sorting  $\mathbf{e}$  by performing the equivalent of updating the permutation matrix  $\mathbf{P}$ . This slow update suggests a steepest descent update formula with small constant update parameter  $\alpha$ . A conjugate gradient approach would be converging way too quickly to ensure moment constraint are reasonably satisfied. In this regard, we could use something like an iterative LMS algorithm [Lage00].

4. The stopping criteria is also important. We want to satisfy the constraints as much as possible so that our solution will make the most sense. We can notice however that our Langrangian has two kinds of competing criteria. That is, in the cost, we want to minimize the energy of our error signal as much as possible. But, in the constraint, we need the error signal to be such that the moments estimated from it match the known noise moments. This suggests that one should continue decreasing the cost until the average relative error between the true and estimated moments starts increasing. If we did not do this, the algorithm would naturally continue minimizing the error energy (the cost  $\mathbf{e}^T \mathbf{e}$ ) even if this meant that the constraints were no longer truly satisfied. This is our objective stopping criteria that we could use to terminate the deconvolution iterations.

## Chapter 5

### Experimental Results

This chapter will present deconvolution results on four different restoration approaches: inverse filtering and Wiener filtering as described in chapter 3 and constrained deconvolution using the standard and PW moments described in chapter 4. The noise constrained approaches will be examined both with and without pixel range constraints. We will show results using the standard “Lena” image for both well behaved and poorly behaved filters and do so over a wide range of signal-to-noise ratios.

#### 5.1 Filters Considered

In order to test a deconvolution algorithm we must have an original image and a degraded version of this original. In this way, the deconvolution algorithm’s ability to restore the degraded image can be directly quantified numerically as well as qualified visually. Our degrading will consist of low pass filtering the original image and then adding noise to the resulting blurred image.

Because we have previously mentioned that the filter’s characteristics influence to what extent an image can be restored, i.e. any nulling of frequency components in the signal cannot be recovered without a priori knowledge of signal content, we will show results using two different “well behaved” filters that “softly” attenuate frequencies without nulling any signal components and two different “poor behaved” filters that do null signal components.

We will assume each of these filters is separable, symmetric and energy preserving. This last criteria means the filter must satisfy

$$\sum_{m_1=-K_1}^{K_1} \sum_{m_2=-K_2}^{K_2} h[m_1, m_2] = 1 \quad (5.1)$$

It is a condition such that the filtered image retains its mean-value, i.e. there is no darkening or lightening of the image as a result of the filtering. The separability of the filter means that the 2-D filter's form can be expressed as the product of two 1-D filters. This is given in eqn. (5.2) in sequence form as

$$h[m_1, m_2] = h_1[m_1]h_2[m_2] \quad (5.2)$$

or equivalently in matrix form as

$$\mathbf{h} = \mathbf{h}_1 \mathbf{h}_2^T \quad (5.3)$$

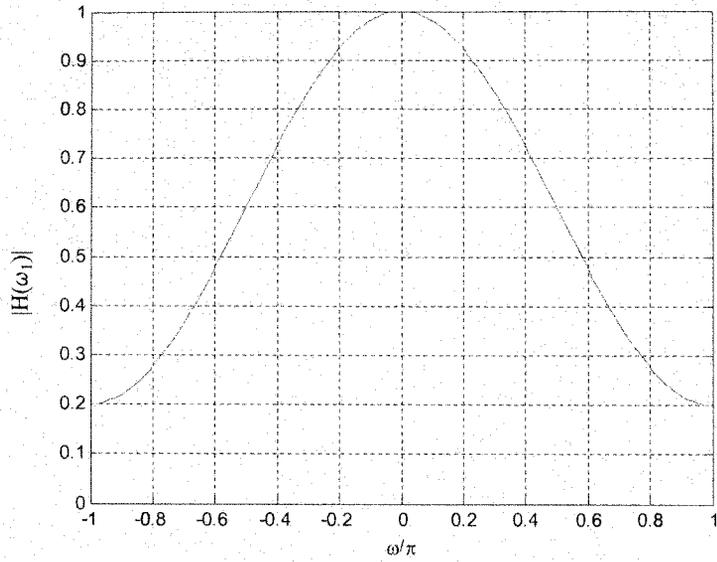
The first “well behaved” filter is symmetric and is based on  $\mathbf{h}_1 = [0.2, 0.6, 0.2]^T$ .

Thus, using eqn. (5.3), we get a filter of

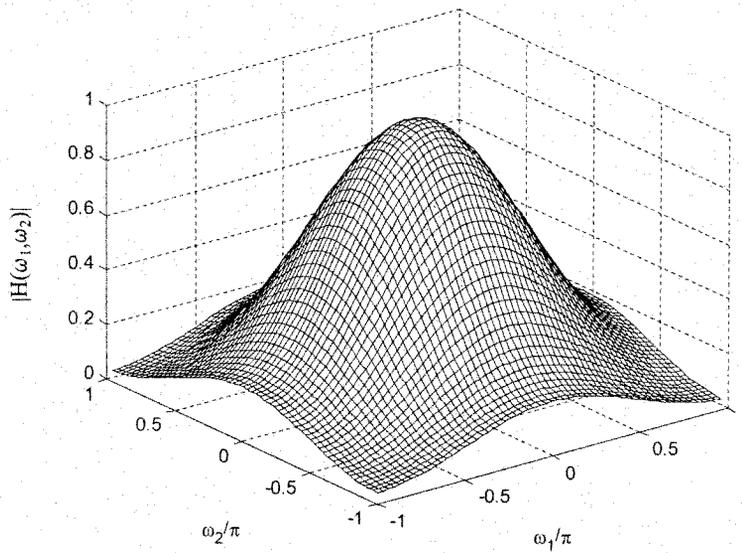
$$\mathbf{h} = \begin{bmatrix} 0.04 & 0.12 & 0.04 \\ 0.12 & 0.36 & 0.12 \\ 0.04 & 0.12 & 0.04 \end{bmatrix} \quad (5.4)$$

Making a spectral plot of this filter, as done in fig (5.1), we can notice that there are no frequency nulls. We have provided the spectral response of  $h_1$  in 1-D to better appreciate the filter's character. Then, in fig (5.1b), we have included the spectral response of the 2-D filter of eqn. (5.4) for completeness.

The second “well behaved” filter is a 5-tap low pass filter. It is given by  $\mathbf{h}_1 = [0.05, 0.2, 0.4, 0.2, 0.05]^T / 0.9$ . It is also symmetric, separable and energy preserving and is given by



(a)



(b)

Figure 5-1. Spectral response of the “well-behaved” low pass filter given by eqn. (5.4). (a) Spectra of  $\mathbf{h} = [0.2, 0.6, 0.2]^T$  upon which eqn. (5.4) is based. (b) Spectra of 2-D filter.

$$\mathbf{h} = \begin{bmatrix} 0.0028 & 0.0111 & 0.0222 & 0.0111 & 0.0028 \\ 0.0111 & 0.0444 & 0.0889 & 0.0444 & 0.0111 \\ 0.0222 & 0.0889 & 0.1778 & 0.0889 & 0.0222 \\ 0.0111 & 0.0444 & 0.0889 & 0.0444 & 0.0111 \\ 0.0028 & 0.0111 & 0.0222 & 0.0111 & 0.0028 \end{bmatrix} \quad (5.5)$$

As with the previous filter example, fig (5.2) illustrates the spectral characteristics of this filter. Notice that there are no frequencies nulled in this filter response.

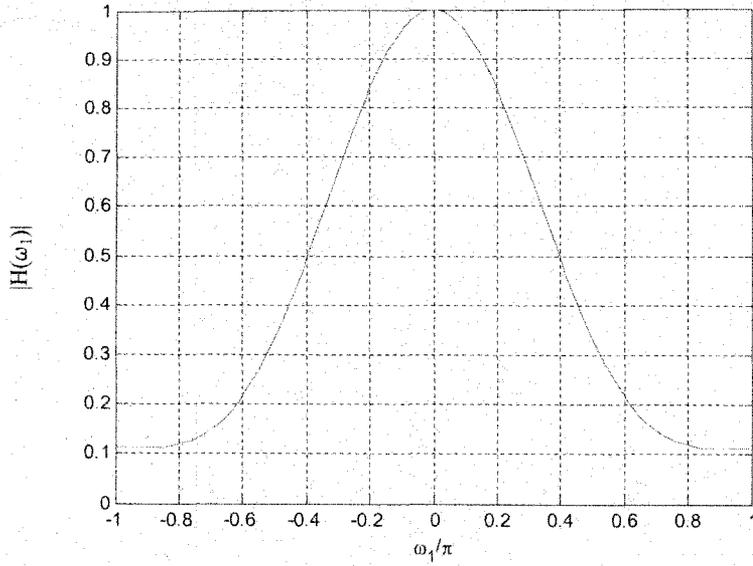
For the “poorly behaved” filters, we will be using moving average (MA) filters. One will be a 3-tap MA filter and the other a 5-tap MA filter. We can notice that these are also symmetric and energy preserving as  $\mathbf{h}_1 = [1,1,1]^r/3$  in one case and  $\mathbf{h}_1 = [1,1,1,1,1]^r/5$  in the other so that

$$\mathbf{h} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \frac{1}{9} \quad (5.6)$$

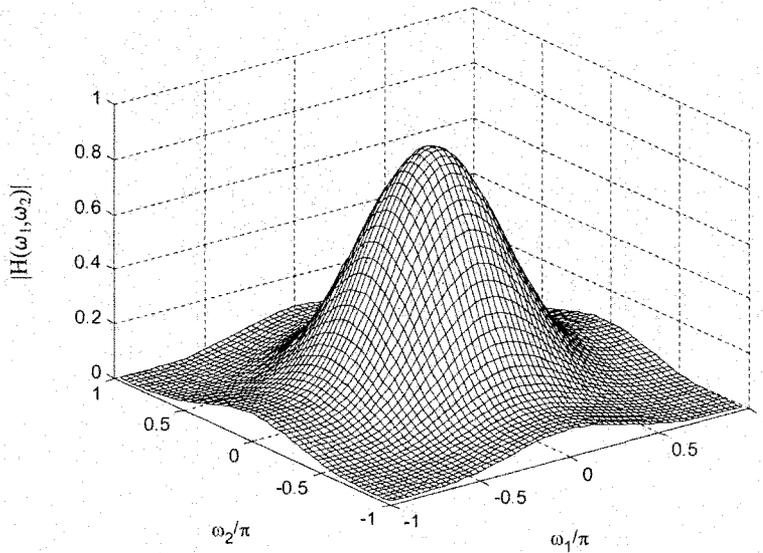
and

$$\mathbf{h} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \frac{1}{25} \quad (5.7)$$

respectively. The “poor” deconvolution characteristics in these filters is clear upon examining their spectral responses. One can clearly see the frequency nulling properties these filters exhibit in figs. (5-3) and (5-4).

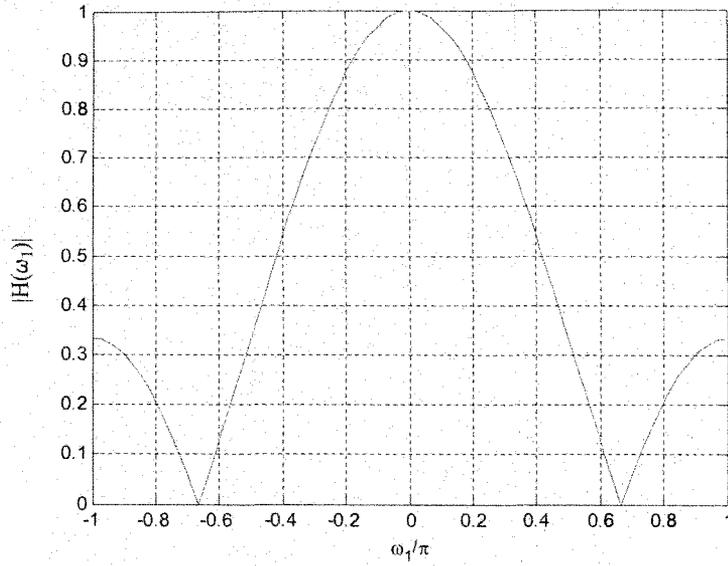


(a)

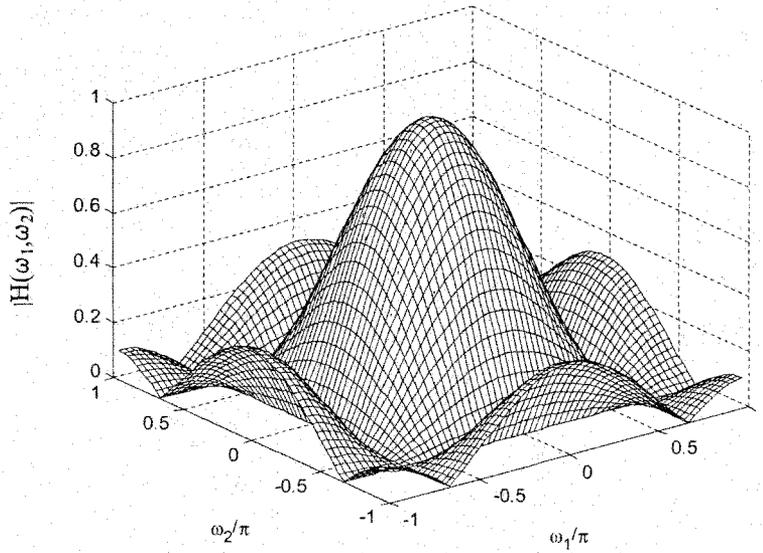


(b)

Figure 5-2. Spectral response of the second “well-behaved” low pass filter given by eqn. (5.5). (a) Spectra of  $\mathbf{h}_1 = [0.05, 0.2, 0.4, 0.2, 0.05]^T / 0.9$  upon which eqn. (5.5) is based. (b) Spectra of 2-D filter.

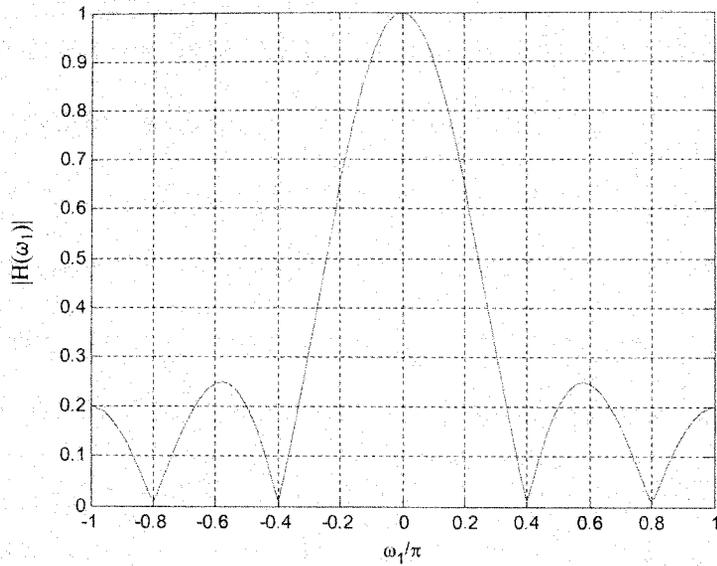


(a)

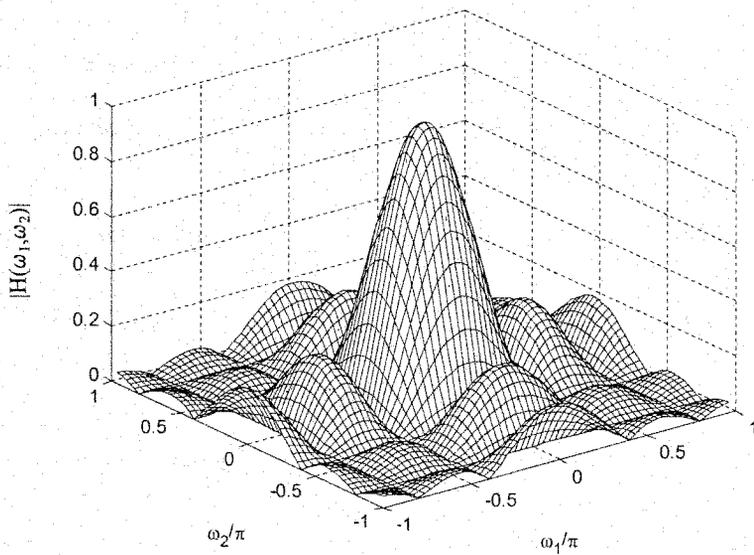


(b)

Figure 5-3. Spectral response of the “poorly” behaved low pass filter given by eqn. (5.6). (a) Spectra of  $\mathbf{h}_1 = [1, 1, 1]^T / 3$  upon which eqn. (5.6) is based. (b) Spectra of 2-D filter.



(a)



(b)

Figure 5-4. Spectral response of the “poorly” behaved low pass filter given by eqn. (5.7). (a) Spectra of  $\mathbf{h}_1 = [1, 1, 1, 1, 1]^T / 5$  upon which eqn. (5.7) is based. (b) Spectra of 2-D filter.

## 5.2 Analysis and Comparison

In this section, we analyze the results of the four deconvolution algorithms we have introduced in this thesis. All our experiments are controlled in that we convolve a known original image with one of the low pass filters described in the previous section. Then, Gaussian noise of known characteristics is added to the convolved original image. The noise added will be conveniently specified in terms of signal-to-noise (SNR) ratio. This is done to give a more intuitive sense of the amount of noise added to an image as different images likely require that the noise variance used to achieve the same SNR be different. In other words, a corrupted image with an SNR of 30 dB is usually slightly visibly noisy and SNRs  $> 40$  dB usually result in images very near the original image with noise barely, if at all, noticeable. In contrast, it is more difficult to “get a feel” of how noisy an image corrupted with a standard deviation of  $\sigma = 3.23$ , for example, is because this specification is not intuitively relative to anything.

### 5.2.1 Quantitative Measures Used

The power of noise added to an image is determined from the desired signal-to-noise ratio. Since noise is added to the blurred image, the signal power must be determined from the blurred image. To determine the appropriate noise power, we use the SNR relation of

$$\text{SNR} = 10 \cdot \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (5.8)$$

Note that because we are adding noise to the blurred image, the SNR we refer to is sometimes known as the blurred signal to noise ratio. We will just refer to it as the SNR with this understanding in mind.

Since we are working with zero mean Gaussian noise, the noise variance needed for generating the appropriate noise samples to add comes from eqn. (5.8). It is seen to be

$$\sigma_N^2 \equiv P_{noise} = \frac{P_{signal}}{10^{SNR/10}} \quad (5.9)$$

where the signal power  $P_{signal}$  is estimated as the variance of the blurred image's samples, that is

$$P_{signal} \approx \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} z^2[n_1, n_2] \right] - \left[ \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} z[n_1, n_2] \right]^2 \quad (5.10)$$

for an image  $z$  of size  $N_1 \times N_2$  samples.

To quantitatively demonstrate an algorithm's performance, we will employ the commonly used measure of peak-signal to noise ratio (PSNR). It is known that this measure does not always correlate reconstruction error with visual quality, but since no such measure has been rigorously established nor does a de facto metric exist, we use PSNR. The definition of PSNR is

$$\text{PSNR} = -10 \log_{10} (e_{rms}^2) \quad (5.11)$$

where

$$e_{rms}^2 = \frac{1}{N_1 N_2} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} (x[n_1, n_2] - \hat{x}[n_1, n_2])^2 \quad (5.12)$$

Notice that  $\hat{x}[n_1, n_2]$  and  $x[n_1, n_2]$  in eqn. (5.12) take values relative to the unit interval  $[0,1]$  according to the pixel normalization utilized in this thesis and that  $x$  refers to the original image while  $\hat{x}$  refers to the restored image.

### 5.2.2 Performance of Methods

In this section, we demonstrate the results of the four deconvolution algorithms under different filtering, noise and constraint conditions. We will work with the “Lena” image, common in the image processing literature, throughout. This image will be shown in the top left portion of the figures along side the restoration image results. The results to be demonstrated make use of the four filters described in section 5.1. We also demonstrate results for deconvolution when either  $M = 1$  or  $M = 3$  noise moments are considered. Please note that several plots and/or images will be presented in this section. This is but a small sampling of all of the possible combinations that could have been considered in analyzing the deconvolution routines’ performance. That is, if you considered running tests over  $N_i$  images,  $N_f$  filters,  $N_m$  moment constraints,  $N_n$  noise condition, etc., you end up with a large number of scenarios for which to analyze and report results. In this thesis, we have elected to present a sufficient set of results that can characterize the performance of the deconvolution methods while not inundating the reader in details. The main results consist of plots of restoration performance over a wide range of degraded image SNR. Specifically, for SNRs from 5 to 100 dB (in increments of 5 dB), we plot the restored images’ PSNR for each of the six approaches considered: Wiener filtering, inverse filtering, PWM optimization with and without range constraints and standard moment constraints with and without range constraints. We now note that

inverse filtering results will not be reported when analyzing the poorly behaved filters as no inverse filter exists in these cases.

To present the results in an organized manner, there will be one figure with plots for each of the four filters considered (and discussed in section 5.1). The ‘(a)’ plot subfigures will correspond to utilizing 1 noise-moment constraint and the ‘(b)’ plot subfigures will show results using 3 noise-moment constraints. Then, following each of the four quantitative (in terms of PSNR) restoration result plots, there are several image based figures that allow the reader to qualitatively inspect the restored images. This is useful as PSNR is not always a faithful indicator of restoration quality for SNRs < 35 dB usually. Note that the image figures following each plot are but a subset of all the images generated when creating the PSNR vs. SNR curves. Also, due to the large amount of figures, all result are found at the end of this chapter.

Both the Wiener and inverse filters were implemented as described in chapter 3. For the Wiener filter, we implemented eqn. (3.40) and we also used the approximation  $S_x(\omega_1, \omega_2) \approx S_y(\omega_1, \omega_2) - \sigma_N^2$  of eqn. (3.43) in the definition. Because we were degrading the images in a controlled environment, we knew what the noise variance  $\sigma^2$  was exactly. This information was used in carrying out the experiments. The inverse filter was only used in those cases where we had a “well behaved” filter, that is the ones of eqns. (5.4) and (5.5). This is why there are no inverse filtering results shown when the filter used was “poorly” behaved. When we do show inverse filtered images, note that the size of the image is smaller than the degraded image being deconvolved. This is because we only kept the “valid” part of the inverse filtered convolution so that we could make a fair comparison of the restored image in terms of PSNR.

In the first set of results, we used the 3-tap “well behaved” filter of eqn. (5.4) for blurring the original “Lena” image. Noise was then added to this image such that SNRs of 5, 10,..., 100 dB were achieved following eqns. (5.8)-(5.10). The PSNR vs. SNR curves for the  $M = 1$  and  $M = 3$  noise constraint cases are given in fig. (5-5a) and (5-5b), respectively. In general, we can see that the noise moment constrained approaches perform better than the inverse-filter-related approaches of the Wiener and inverse filters for SNRs  $> 15$  dB. It is interesting to note that the standard noise moment constraints approach performs very well for SNRs  $> 30$  dB. In contrast, the PWM approach performs better than the inverse filtering and standard moment approaches for the  $15 \text{ dB} < \text{SNR} < 22 \text{ dB}$  range. That is, in clean (very little noise images), the standard approach is preferred. For substantial noise in an image, the PWM approach is preferred. But when the image is saturated with noise, the Wiener filter might provide the best trade off (at least visually) between smoothing the image while suppressing noise. These effects can be seen visually by inspecting the images of figs. (5-6)-(5.11). In each of these figures, we illustrate the original and degraded (“Blurred + Noise”) images as well as the probability weighted moment constrained with range (“PWM-with range”) and without range (“PWM-no range”) restored images, the standard noise moment constrained with range (“SNM-with range”) and without range (“SNM-no range”) restored images and the Wiener and inverse filtered images. In these images we can see that the inverse filtered images amplify noise unacceptably at low SNRs ( $< 15$  dB). The Wiener filtered images trade-off image sharpness (restoration) for noise attenuation at low SNRs. The standard noise moment images do the best job of removing the blurring effect but do not handle noise as well as the PWM approach does. That is, the PWM images “control” the noise

variance to its proper level but do not show unblurring results as effective as the SNM images do. Finally, we can see in the plots that using  $M = 1$  and  $M = 3$  noise constraints did not affect the results significantly.

In the next case analyzed, a 5-tap well behaved filter was used. The plots of fig. (5-12) demonstrate similar behavior to that of the “well behaved” 3-tap filter case but the Wiener filtering performance is now only superior in PSNR to the noise constrained approaches when  $\text{SNR} < 10$  dB. Otherwise, the performances of all methods are very similar to that shown previously. What we see different in this scenario is that, due to the greater blurring on the original image, it is a bit more difficult to remove the blurring and recover a sharp, restored image in this case relative to the 3-tap case shown before. The images resulting from the 5-tap case (corresponding to a subset of the results of the plots in fig. (5-12)) are given in figs. (5-13)-(5-18). In the “well-behaved” filter case, all of the approaches perform fairly well when the noise level is reasonably low. The Wiener filter experiences border effects and this is partly the reason for the lower PSNR that is shown in the plots relative to the other approaches. As previously explained, that is a common problem with direct inverse filtering based approaches. These “ringing” border effects are not present in the noise moment constrained approaches.

The next pair of plots will illustrate deconvolution performance when the blurring function is “poorly-behaved” in the sense of deconvolution processing. These filters necessarily have frequencies nulled in their spectral responses. The first of these sets of results uses the “poorly-behaved” 3-tap moving average filter of eqn. (5.6). The plots are shown in fig. (5-19). Again, Wiener filter performance surpasses all others’ in PSNR for  $\text{SNR} < 10$  dB approximately. There is similar behavior in the moment constrained

approaches where the PWM approach controls noise better in the  $15 \text{ dB} < \text{SNR} < 25 \text{ dB}$  range approximately but, after the noise level becomes small enough, the deblurring ability of the SNM approach produces a sharper image than the PWM case. Note that the routines cannot restore a perfectly sharp image since the frequency nulling properties of the filter prevent certain spatial frequencies from being recovered. Without a priori knowledge of the original image, these spatial frequencies could never be recovered automatically.

The image resulting from the deconvolution procedures are pictured in fig. (5-20) through (5-25). The PWM approach produces images just as sharp, if not sharper, than the Wiener filtering images. However, we can notice the “ringing” behavior in the Wiener images becomes more pronounced as the noise variance decreased (SNR increases). This ringing border effect becomes unacceptable at high SNRs. The SNM approach produces the sharpest images at high SNRs. It has not been mentioned until now, but the range constraining does not have much of an effect in the moment constrained deconvolution approaches when the  $\text{SNR} > 10 \text{ dB}$ . It seems only in very low SNR cases does this have much of an effect.

The final series of results make use of the “poorly-behaved” 5-tap MA filter of eqn. (5.7). Again, behavior performance follows the same trend as that previously described. Notice that the 5-tap MA filter is more aggressively filtering the image relative to the other filtering that has been performed. There are more frequency nulls in this filter too. The PSNR vs. SNR plots are given in fig. (5-26) for this case. Some image results corresponding to these plots are provided in fig. (5-27) to (5-32). In general, it is more

difficult to restore the degraded image in this case due to the relatively poor characteristics of this filter compared to the others.

Given all the results we have presented, we can conclude the following regarding these deconvolution approaches:

- Deconvolution becomes increasingly more difficult, in general, as the amount of frequency nulling increases.
- Deconvolution becomes increasingly more difficult, in general, as the SNR of the degraded image decreases.
- Direct inverse filtering is useful with well-behaved filters (no frequency nulls) when the SNR is high.
- Wiener filtering is most useful with well behaved filters or at low SNR with poorly behaved filters. In either case, the Wiener filter is highly susceptible to “ringing” effects at the borders of images. This type of filtering is usually used to trade off filter uninvertibility with smoothing of the results. In essence, this type of operation just uses a smoother approximation to the inverse filter.
- The noise moment constrained approaches are most effective at higher SNRs. They do not suffer from ringing effects at the borders and are useful at maintaining a given noise level. Rather than trading off smoothness with noise attenuation, they attempt to unblur the image while not amplifying noise. This non-amplification effect is more effective when using PWM moments relative to standard moments but the standard moments do a better job of correcting for the blurring that has occurred.

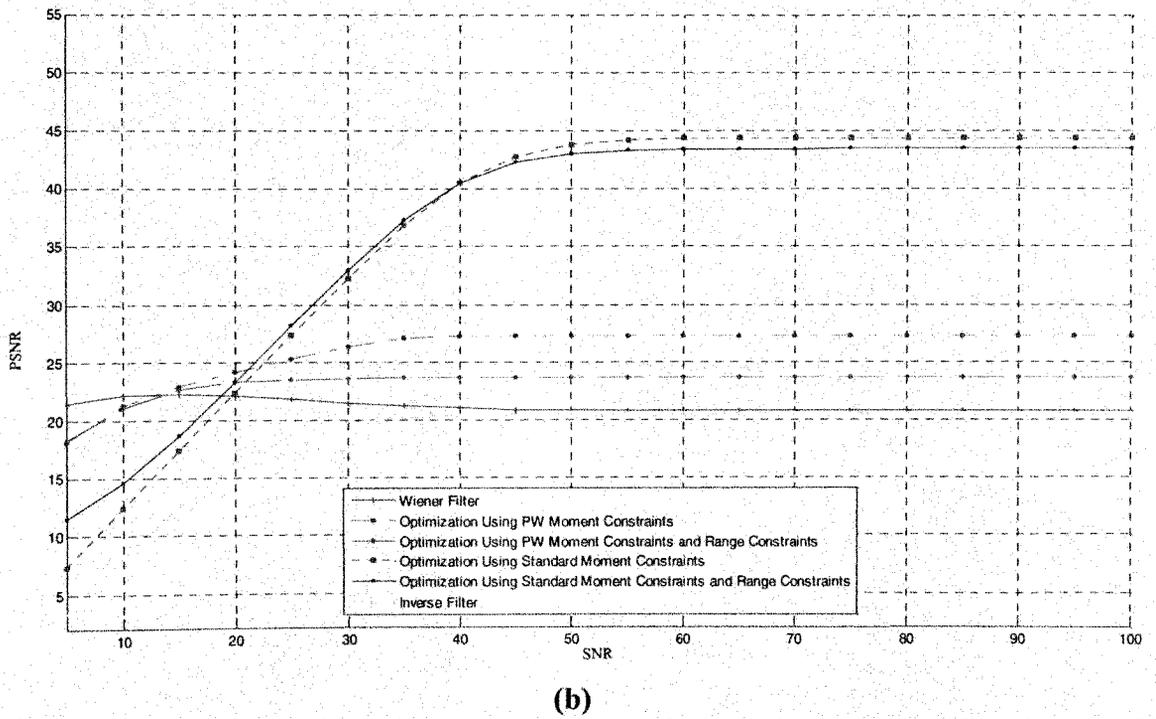
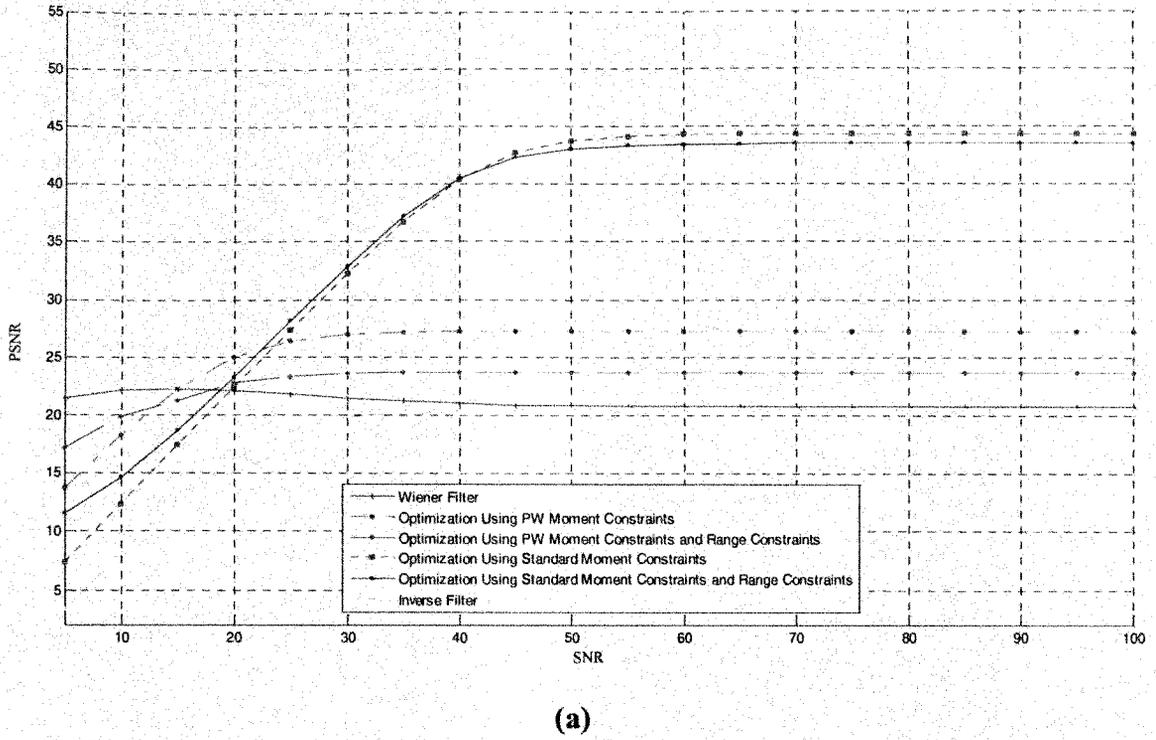


Figure 5-5. Results with the 3-tap “well-behaved” filter where  $h_1 = [0.2, 0.6, 0.2]^T$ .  
 (a) Curves using  $M=1$  noise moment (b) curves using  $M=3$  noise moments.

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



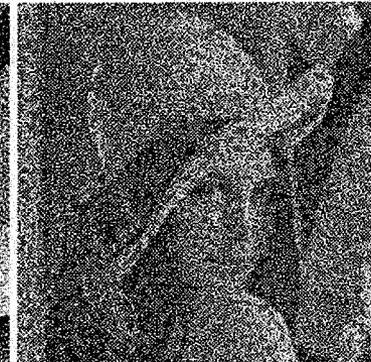
**PWM-with range**



**SNM-with range**



**Inverse**



**Figure 5-6. Restoration results corresponding to the plot in fig. (5-5a) when SNR=5dB and  $M=1$ .**

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



**Inverse**



**Figure 5-7. Restoration results corresponding to the plot in fig. (5-5a) when SNR=15dB and  $M=1$ .**

Original



PWM-no range



SNM-no range



Wiener



Blurred + Noise



PWM-with range



SNM-with range



Inverse



Figure 5-8. Restoration results corresponding to the plot in fig. (5-5a) when SNR=25dB and  $M=1$ .

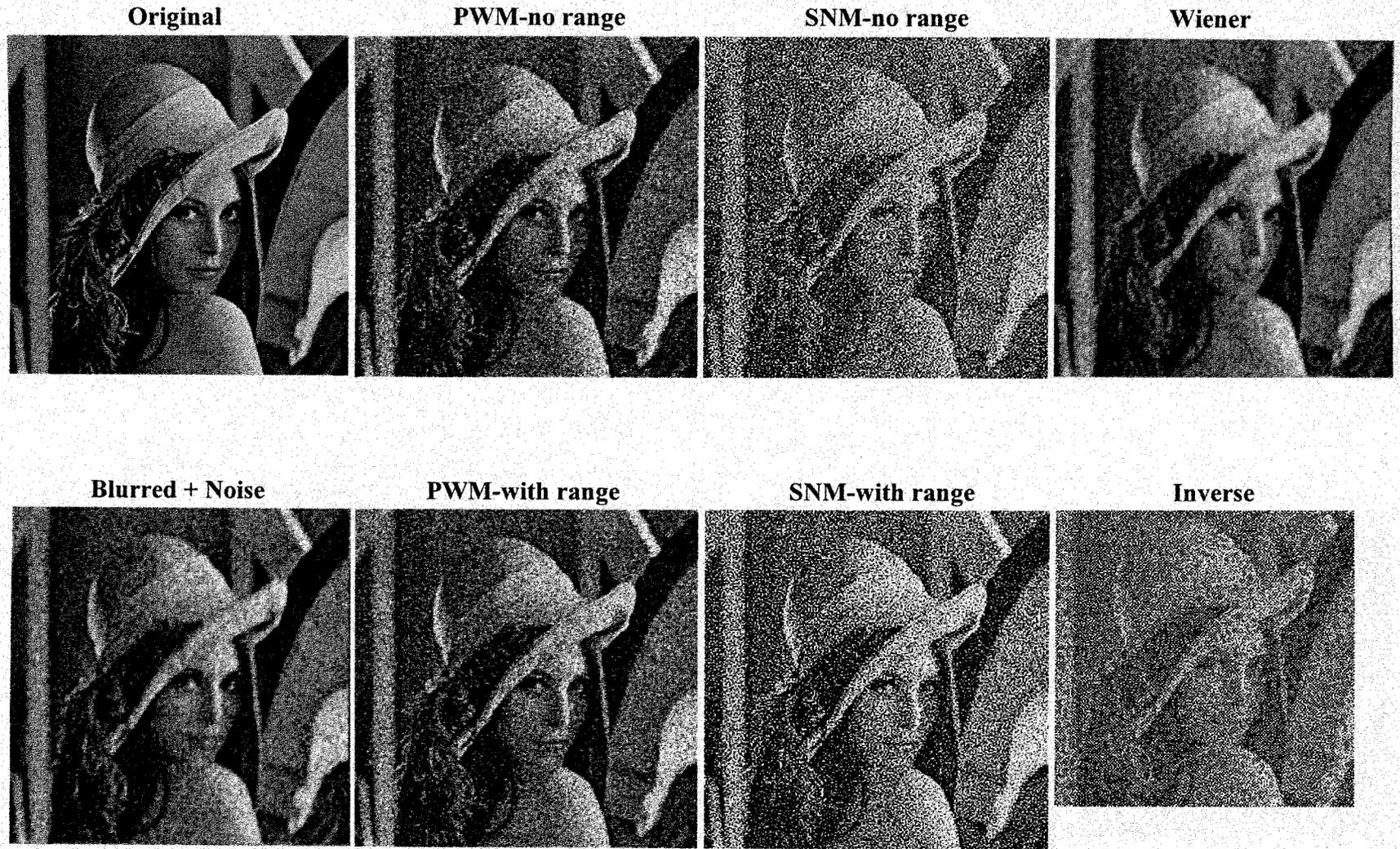


Figure 5-9. Restoration results corresponding to the plot in fig. (5-5b) when SNR=5dB and  $M=3$ .



Figure 5-10. Restoration results corresponding to the plot in fig. (5-5b) when SNR=15dB and  $M=3$ .

Original



PWM-no range



SNM-no range



Wiener



Blurred + Noise



PWM-with range



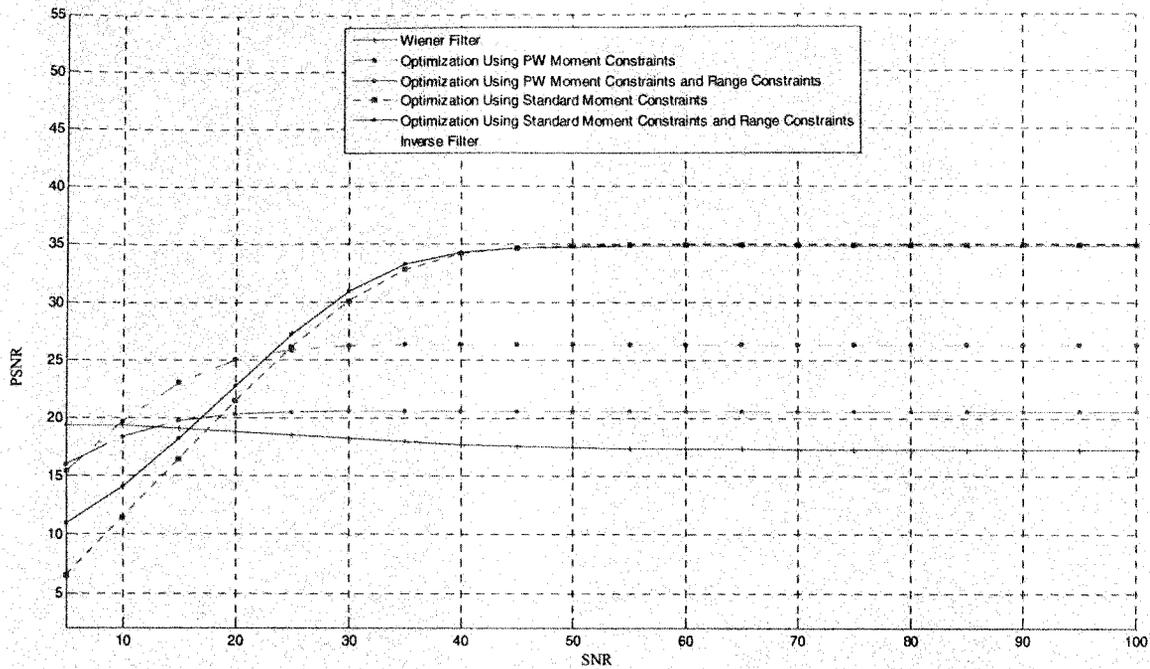
SNM-with range



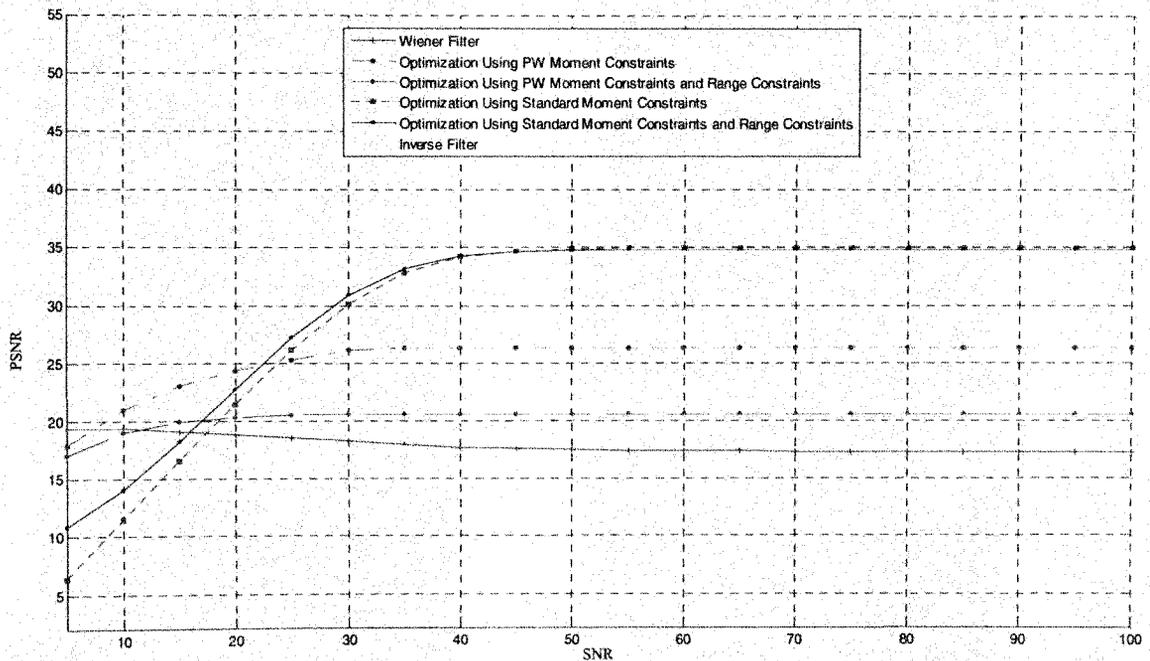
Inverse



Figure 5-11. Restoration results corresponding to the plot in fig. (5-5b) when SNR=25dB and  $M=3$ .



(a)



(b)

Figure 5-12. Results with the 5-tap “well-behaved” filter where  $\mathbf{h}_1 = [0.05, 0.2, 0.4, 0.2, 0.05]^T / 0.9$ . (a) Curves using  $M=1$  noise moment (b) curves using  $M=3$  noise moments.



Figure 5-13. Restoration results corresponding to the plot in fig. (5-12a) when SNR=15dB and  $M=1$ .



Figure 5-14. Restoration results corresponding to the plot in fig. (5-12a) when SNR=25dB and  $M=1$ .

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



**Inverse**



**Figure 5-15.** Restoration results corresponding to the plot in fig. (5-12a) when SNR=35dB and  $M=1$ .



Figure 5-16. Restoration results corresponding to the plot in fig. (5-12b) when SNR=15dB and  $M=3$ .

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



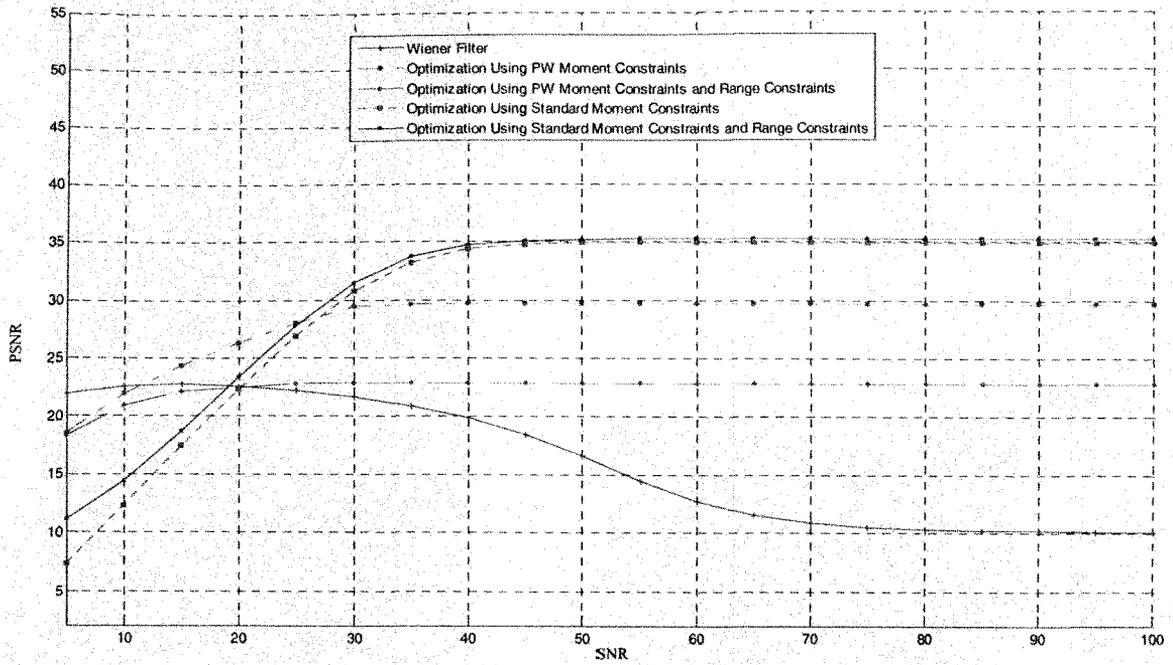
**Inverse**



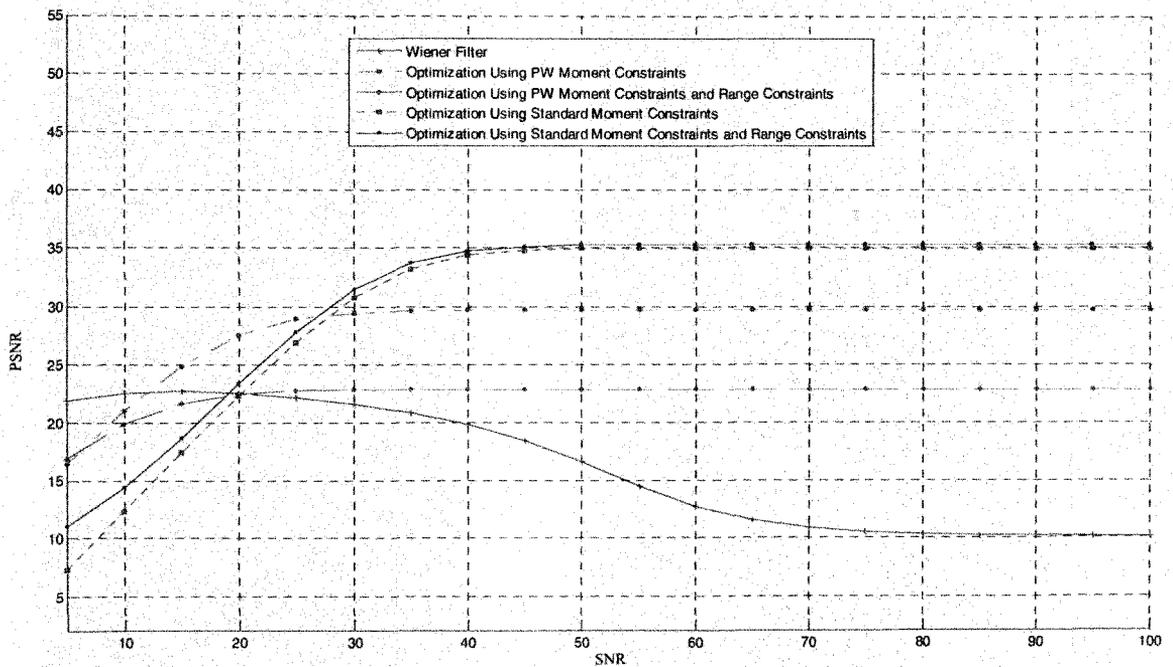
**Figure 5-17. Restoration results corresponding to the plot in fig. (5-12b) when SNR=25dB and  $M=3$ .**



Figure 5-18. Restoration results corresponding to the plot in fig. (5-12b) when SNR=35dB and  $M=3$



(a)



(b)

Figure 5-19. Results with the 3-tap “poorly behaved” filter where  $\mathbf{h}_1 = [1, 1, 1]^T / 3$ .  
 (a) Curves using  $M=1$  noise moment (b) curves using  $M=3$  noise moments.

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



**Figure 5-20.** Restoration results corresponding to the plot in fig. (5-19a) when SNR=15dB and  $M=1$ .



Figure 5-21. Restoration results corresponding to the plot in fig. (5-19a) when SNR=25dB and  $M=1$ .

**Original**

**PWM-no range**

**SNM-no range**

**Wiener**



**Blurred + Noise**

**PWM-with range**

**SNM-with range**



**Figure 5-22. Restoration results corresponding to the plot in fig. (5-19a) when SNR=35dB and  $M=1$ .**



Figure 5-23. Restoration results corresponding to the plot in fig. (5-19b) when SNR=15dB and  $M=3$ .

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



**Figure 5-24. Restoration results corresponding to the plot in fig. (5-19b) when SNR=25dB and  $M=3$ .**

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



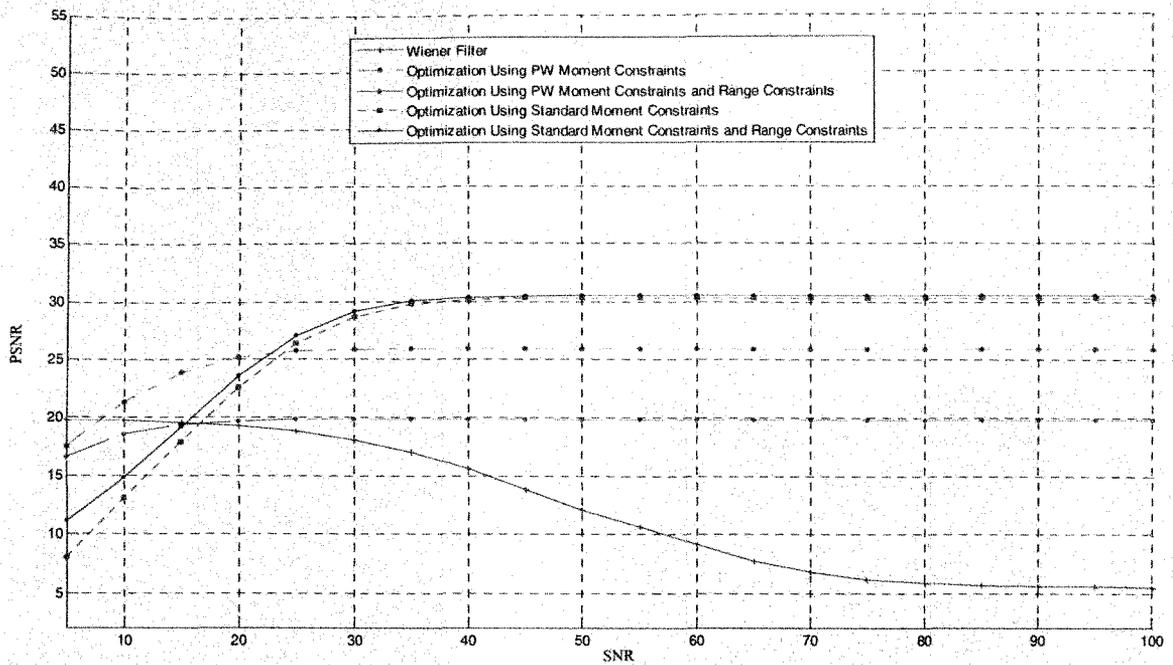
**PWM-with range**



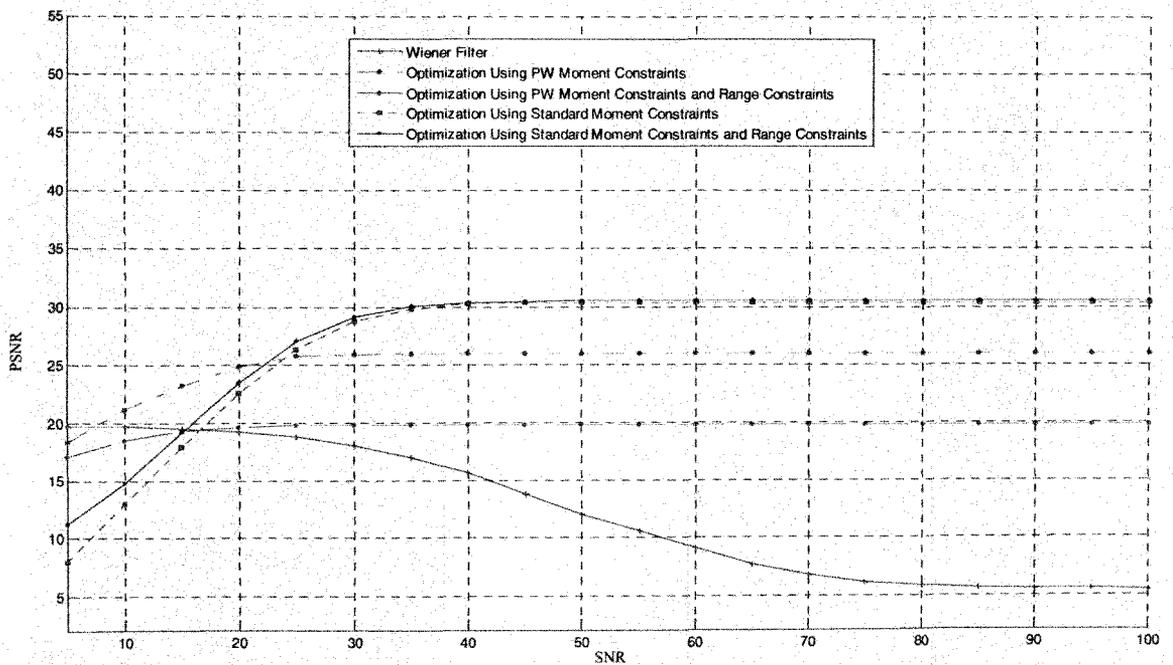
**SNM-with range**



**Figure 5-25.** Restoration results corresponding to the plot in fig. (5-19b) when SNR=35dB and  $M=3$ .



(a)



(b)

Figure 5-26. Results with the 5-tap “poorly behaved” filter where  $\mathbf{h}_1 = [1, 1, 1, 1, 1]^T / 5$ . (a) Curves using  $M=1$  noise moment (b) curves using  $M=3$  noise moments.

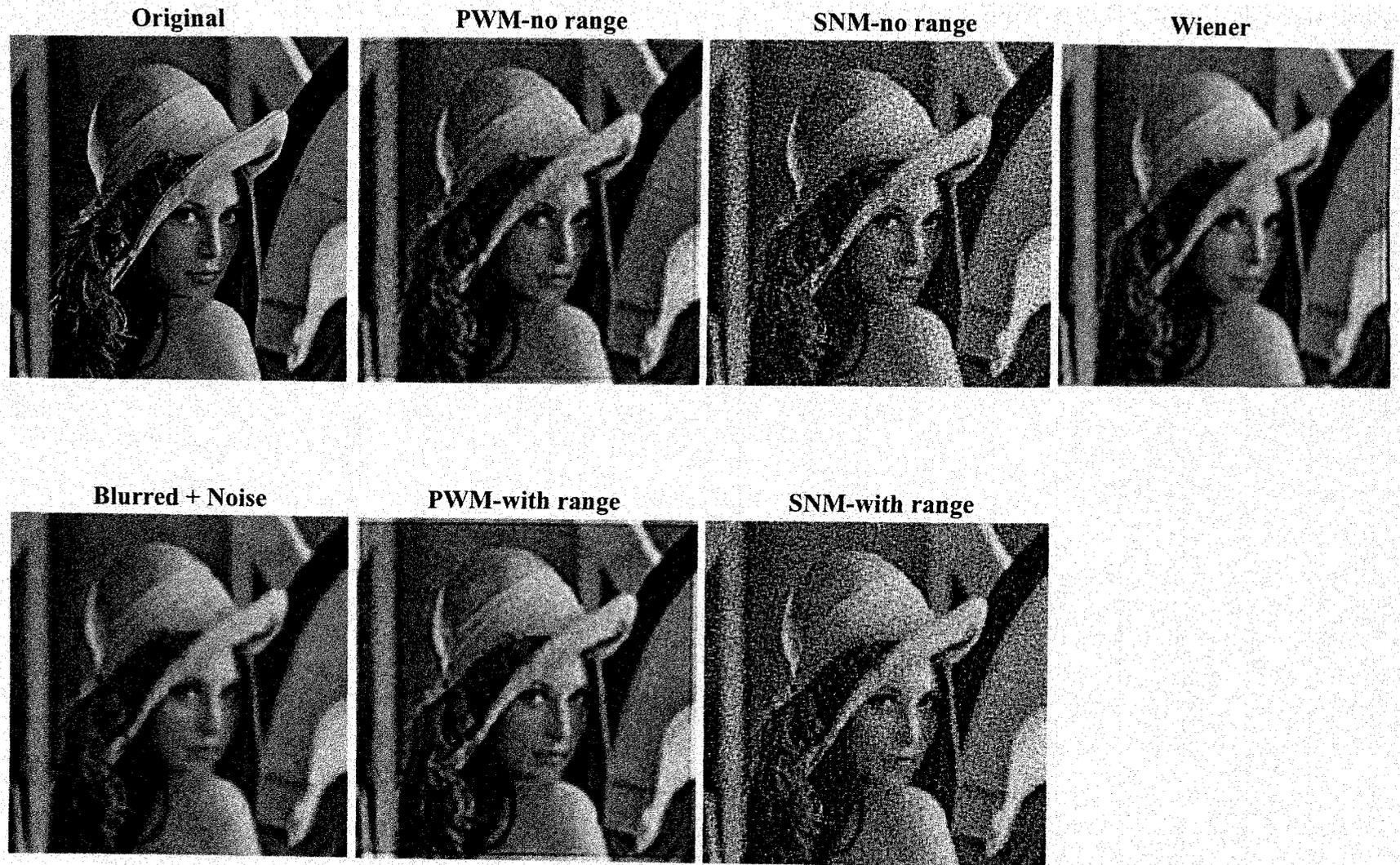


Figure 5-27. Restoration results corresponding to the plot in fig. (5-26a) when SNR=15dB and  $M=1$ .

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



**Figure 5-28.** Restoration results corresponding to the plot in fig. (5-26a) when SNR=25dB and  $M=1$ .

**Original**



**PWM-no range**



**SNM-no range**



**Wiener**



**Blurred + Noise**



**PWM-with range**



**SNM-with range**



**Figure 5-29. Restoration results corresponding to the plot in fig. (5-26a) when SNR=35dB and  $M=1$ .**



Figure 5-30. Restoration results corresponding to the plot in fig. (5-26b) when SNR=15dB and  $M=3$ .

Original



PWM-no range



SNM-no range



Wiener



Blurred + Noise



PWM-with range



SNM-with range



Figure 5-31. Restoration results corresponding to the plot in fig. (5-26b) when SNR=25dB and  $M=3$ .



Figure 5-32. Restoration results corresponding to the plot in fig. (5-26b) when SNR=35dB and  $M=3$ .

## Chapter 6

### Conclusion and Future Work

In this thesis we have proposed an optimally constraining image deconvolution algorithm employing range and noise moment constraints. This algorithm uses a non-linear variable transformation to simplify the deconvolution solution and to enforce range-limiting data constraints. It relies on prior knowledge of the statistics of the additive noise to ensure that the error signal matches the true moments of the noise distribution. Two types of noise moment constraints on the additive noise were studied in this thesis: standard moments and probability weighted moments. Standard noise moment definitions are common in probabilistic analysis but they require a nonlinear optimization method when used in deconvolution. Because of this, we proposed using probability weighted moments in our noise constraints since they simplify the mathematical complexity related with moment estimation computation and the deconvolution procedure in general. According to our analysis, the use of each type of moment has its advantages depending on the SNR of the degraded image.

The analysis of the deconvolution methods was carried out quantitatively by plotting the peak signal to noise ratio of the recovered images vs. the signal to noise ratio used in each restoration scenario. From the results, we observed that the noise moment constrained algorithms do not suffer from “ringing” effects at the border like the Wiener filter does. This is clearly evident in the case of high SNR with “poorly” behaving blurring filters, i.e. those that null one or more frequencies. The noise constrained approaches, unlike the Wiener filter, do not trade-off noise amplification for smoothing.

Rather, they try to maintain a given noise level while unblurring the blurred image. This is particularly effective at high SNR.

In our analysis, we found that incorporating range constraints in the deconvolution operation while enforcing noise moment constraints was not of much benefit at higher SNRs. It was most beneficial at very low SNRs. Also, we found that good deconvolution results could be achieved with as little as using one noise moment. Indeed, the differences between using 1 and 3 moment constraints were very little. The use of noise moments was found to be both novel and effective to performing deconvolution.

### *Future Work*

The field of deconvolution has a long history. And, given all the work that has been performed, it is clear that constraints must somehow be employed in obtaining an acceptable restored image. In this thesis, we have brought to light the difference between attempting to deconvolve with a well behaved filter and a poorly behaved one (in the sense of deconvolution processing). When dealing with well behaved filters, the focus is on dealing with the ill-effects of noise. Further study on how effective noise moment constraints are with different types of noise is an interesting topic that was beyond the scope of this thesis. When dealing with poorly behaved filters, the deconvolution approach must not only consider the effects of noise but also how this, coupled with unrecoverable spatial frequency content, can best be addressed so that an acceptable image can be produced. It is not clear presently how reliably noise moment constraints deal with this problem. By solving this issue, one would positively influence the future directions and efforts of deconvolution-based research.

## List of References

- [Axel00] N. Axelrod, A. Radko, N. Ben-Yosef, A. Khatchatouriants, M. Treinin, and A. Lewis, "Near-field optical and atomic force constraints for superresolution 3D deconvolution in far-field optical microscopy," in *Three-Dimensional and Multidimensional Microscopy: Image Acquisition Processing VII*, J. A. Conchello, C. J. Cogswell, A. G. Tescher, and T. Wilson, eds., Proc. SPIE 3919, 161–171 2000.
- [Bert95] D. P. Bertsekas, "Nonlinear Programming" (Belmont, MA: Athena Scientific) 1995.
- [Cand04] Candocia, F.M., Svedberg, E., Litvinov, D. and Khizroev, S., "Deconvolution Processing for Increasing the Resolution of Magnetic Force Microscopy Measurements," *Nanotechnology, Special Issue on Nanoscale Devices and Systems Integration*, Vol. 15, No. 10, pp. S575-S584, October 2004.
- [Figu03] Figueiredo, M.A.T., Nowak, R.D., "An EM algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, Vol. 12 Issue 8, August 2003.
- [Gree79] Greenwood, J.A., Landwehr, J.M., Matalas, N.C. & Wallis, J.R. (1979) Probability weighted moments: definition and relation to parameters of several distributions expressible in inverse form. *Wat. Resour. Res.*, 15, 1049-1054
- [Hare00] Hare, J.R., Reilly, J.P., "The Iterative Deconvolution of Linearly Blurred Images using Non-Parametric Stabilizing Function," *IEEE Transactions on Image Processing*, Vol. III, August 2000.
- [Hosk96] Hosking, J. R. M., "Research Report: Fortran routines for use with the method of L-moments". Version 3. IBM Research Division, August 1996.
- [Hunt73] Hunt, B.R., "The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer", *IEEE Transactions on Computers*, vol. 2, pp. 805-812, Sept. 1973.
- [Jalo02] Jalobeanu, A., Nowak, R., Zerubia, J., Figueiredo, M.A.T., " Satellite and aerial image deconvolution using an EM method with complex wavelets," *Proc IEEE International Conf. on Image Processing - ICIP*, Rochester, United States, Vol. 1 , pp. 333 - 336 , September , 2002 .

- [John04] Johnstone, I.M., Kerkyacharian, G., "Wavelet deconvolution using in a periodic setting," Journal of the Royal Statistical Society Series B, Vol. 66 Issue 4 Page 974, November 2004.
- [Kats84] A.K. Katsaggelos, J. Biemond, R.M. Mersereau, R.W. Schafer, "An Iterative Method for Restoring Noisy Blurred Images," Proc. 1984 Int. Conf. Acoust., Speech, Signal Processing, pp. 37.2.1-37.2.4, San Diego, CA, March 1984.
- [Kats91] A.K. Katsaggelos, ed., "Digital Image Restoration," New York: Springer-Verlag, 1991.
- [Lage00] Lagendijk, R.L., Biemond, J., "Basic Methods for Image restoration and Identification," in "Handbook of image and video processing," Al Bovik, Academic Press, San Diego, 2000.
- [Land79] Landwehr, J.M., Matalas, N.C. and Wallis, J.R., 1979, "Probability Weighted Moments Compared with Some Traditional Techniques in Estimating Gumbel Parameters and Quantiles," Water Resources Research, 15(5), pp. 1055-1064.
- [Lim90] Lim, J.S., Two-Dimensional Signal and Image Processing, Englewood Cliffs, NJ, Prentice Hall, 1990, pp. 536-540.
- [Nand03] Nanda, S., "Pre and Post Image Processing for Enhanced Image Compression," Master's Thesis, Florida International University, Fall 2003.
- [Neel99] R. Neelamani, H. Choi, and R. G. Baraniuk, "Wavelet-based Deconvolution using Optimally Regularized Inversion for Ill-Conditioned Systems," SPIE Technical Conference on Wavelet Applications in Signal Processing VII, Vol. 3813, Denver, July 1999.
- [Papo02] Papoulis A and Pillai S U 2002 Probability, Random Variables and Stochastic Processes 4th edn (New York: McGraw-Hill)
- [Sava01] Savage, G.T., Whalen, T.M., and Jeong, G.D., "Application of the self-determined probability-weighted moment method to extreme wind speed estimation," Proceedings of the Americas Conference on Wind Engineering -- 2001, June 4-6, 2001.
- [Star02] J.L. Starck, E. Pantin, Deconvolution in Astronomy: A Review," Pub. Astron. Soc. Pac., 114, 1051-1069, 2002.

## Appendix A. Minimum Norm Solution of Underdetermined Linear Equations

The problem is to solve the constrained quadratic program

$$J = \mathbf{x}^T \mathbf{x} \text{ subject to } \mathbf{A} \mathbf{x} = \mathbf{b} \quad (\text{A.1})$$

where  $\mathbf{A}$  is  $M \times N$  with  $M < N$ . To solve for  $\mathbf{x}$ , we will establish a Lagrangian function as

$$L = \mathbf{x}^T \mathbf{x} + \boldsymbol{\lambda}^T (\mathbf{A} \mathbf{x} - \mathbf{b}) \quad (\text{A.2})$$

Having established the form of our Lagrangian function an optimal minimum solution is found when the vector pair  $\{\boldsymbol{\lambda}, \mathbf{x}\}$  minimizes  $L$ . This requires that the gradient of  $L$  with respect to both  $\mathbf{x}$  and  $\boldsymbol{\lambda}$  be zero, i.e.  $\nabla_{\mathbf{x}} L = \mathbf{0}$  and  $\nabla_{\boldsymbol{\lambda}} L = \mathbf{0}$ , or

$$\nabla_{\mathbf{x}} L = 2\mathbf{x} + \mathbf{A}^T \boldsymbol{\lambda} = \mathbf{0} \quad (\text{A.3})$$

and

$$\nabla_{\boldsymbol{\lambda}} L = \mathbf{A} \mathbf{x} - \mathbf{b} = \mathbf{0} \quad (\text{A.4})$$

Solving for  $\mathbf{x}$ , we first multiply both sides of eqn. (A.3) by  $\mathbf{A}$  to yield an expression for  $\boldsymbol{\lambda}$  as

$$2\mathbf{A} \mathbf{x} + \mathbf{A} \mathbf{A}^T \boldsymbol{\lambda} = \mathbf{0} \quad (\text{A.5})$$

Eqn. (A.5) is equivalent to

$$\boldsymbol{\lambda} = -2(\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{A} \mathbf{x} \quad (\text{A.6})$$

By substituting  $\mathbf{A} \mathbf{x} = \mathbf{b}$  in eqn. (A.6), it can be expressed as

$$\boldsymbol{\lambda} = -2(\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} \quad (\text{A.7})$$

To solve for  $\mathbf{x}$ , we substitute eqn. (A.7) into eqn. (A.3) to get

$$2\mathbf{x} - 2\mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} = \mathbf{0} \quad (\text{A.8})$$

Finally, the solution of  $\mathbf{x}$ , which is the minimum norm solution, is

$$\mathbf{x} = \mathbf{A}^T (\mathbf{A} \mathbf{A}^T)^{-1} \mathbf{b} \quad (\text{A.9})$$

## Appendix B. Deriving Gaussian PW Moments for $X \sim N(0,1)$

To find the PW moments of a general normal distribution, we will first define useful probability moments. This is done because, as shown in [Hosk96], we can generate Gaussian PW moments from these other more easily tabulated moments.

### Z – Moments

The Z-moments are quantities useful in the summation and estimation of probability distributions [Gree79]. The first 20 Z-moments are tabulated below:

**Table B-1. First 20 Z – moments**

$z_1$	0
$z_2$	0.564189583547756287
$z_3$	0
$z_4$	0.122601719540890947
$z_5$	0
$z_6$	0.436611538950024944e-1
$z_7$	0
$z_8$	0.218431360332508776e-1
$z_9$	0
$z_{10}$	0.129635015801507746e-1
$z_{11}$	0
$z_{12}$	0.852962124191705402e-2
$z_{13}$	0
$z_{14}$	0.601389015179323333e-2
$z_{15}$	0
$z_{16}$	0.445558258647650150e-2
$z_{17}$	0
$z_{18}$	0.342643243578076985e-2
$z_{19}$	0
$z_{20}$	0.271267963048139365e-2

Note that all odd-order Z-moments are 0 and that the even Z-moments have been provided out to 20 digit precision for accurate PW moment computation.

## X – Moments

The Z-moments are related to X-moments as follow

$$\begin{aligned}x_1 &= \mu \\x_2 &= \sigma \cdot z_2 \\x_n &= z_n \quad \text{for } n \geq 3\end{aligned}\tag{B.1}$$

## L- Moments

L-moments are certain linear combinations of probability weighted moments. The L-moments are related to the x-moments as

$$\begin{aligned}\lambda_1 &= x_1 \\ \lambda_2 &= x_2 \\ \lambda_n &= \lambda_2 \cdot x_n \quad \text{for } n \geq 3\end{aligned}\tag{B.2}$$

Note that the use of  $\lambda$  has nothing to do with Lagrange multipliers here. We use this notation as it is commonly encountered in the literature.

## PW Moments

Finally, the PW moments are related to L-moments by the following equation

$$b_{m-1} = Q_1 \lambda_1 + \dots + Q_m \lambda_m \quad m = 1, \dots, n\tag{B.3}$$

where  $Q_i$  represents a weighting coefficient. One can notice that the  $\mu$  and  $\sigma$  parameters of a Gaussian distribution are directly related to the Z-moments as in eqn. (B.1).

Using Table B-1 and eqns. (B.1) and (B.3), finding the  $n$  PW moments of a Gaussian distribution requires solving the following linear system of equations:

$$\mathbf{L} \mathbf{b} = \boldsymbol{\lambda}\tag{B.4}$$

where  $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$  and the  $\lambda_i$  values are obtained in eqn. (B.2),  $\mathbf{b} = [b_0, b_1, \dots, b_{n-1}]$  is the vector of PW moments to solve for and  $\mathbf{L}$  is an  $n \times n$  matrix of values that comes from the coefficients of shifted Legendre polynomials. We establish the elements in matrix  $\mathbf{L}$  as described in Algorithm B-1.

Algorithm B-1:

```

 $\mathbf{L}[1,1] = 1$ 
for  $r = 1, \dots, n - 1$ 
  for  $k = 0, \dots, r$ 
     $S = (-1)^{(r-k)}$ 
     $p = 1$ 
    if  $k > 0$ 
      for  $l = 0, \dots, k - 1$ 
         $p = p \cdot \frac{(r - k + 1 + 2 \cdot l)(r - k + 1 + 2 \cdot (l + 1))}{(l + 1)^2}$ 
      end
    end
     $\mathbf{L}[r + 1, k + 1] = p \cdot S$ 
  end
end

```

Thus, the first  $n$  PW moments of a Gaussian distribution is obtained by solving  $\mathbf{b} = \mathbf{L}^{-1}\lambda$  where  $\mathbf{L}$  is always invertible (although it becomes increasingly sensitive to machine precision as  $n$  increases). Note also that  $\mathbf{L}$  is a lower triangular matrix so any values not indexed in matrix  $\mathbf{L}$  in Algorithm B-1 means that those entries in  $\mathbf{L}$  are supposed to be 0.

## Appendix C. Derivative of the Range Mapping Function

The expression of  $x$  as resulting from the nonlinear mapping  $f$  on an auxiliary variable  $\chi$  is given by

$$x = f(\chi) \quad (\text{C.1})$$

where  $f$  is equal to

$$f(\chi) = \frac{1}{2} \tanh(\chi) + \frac{1}{2} \quad (\text{C.2})$$

To demonstrate that the derivative of the nonlinear mapping has the form of

$$f'(\chi) = 2 \cdot f(\chi) \cdot (1 - f(\chi)) \quad (\text{C.3})$$

we square eqn. (C.2), that is

$$f^2(\chi) = \frac{1}{4} \cdot \tanh^2(\chi) + \frac{1}{2} \cdot \tanh(\chi) + \frac{1}{4} \quad (\text{C.4})$$

It is simple to see that eqn. (C.4) can be re-expressed as

$$\begin{aligned} f^2(\chi) &= \frac{1}{4} \cdot \tanh^2(\chi) + \underbrace{\frac{1}{2} \cdot \tanh(\chi)}_{f(\chi)} + \frac{1}{2} - \frac{1}{4} \\ &= \frac{1}{4} \cdot \tanh^2(\chi) + f(\chi) - \frac{1}{4} \\ &= f(\chi) + \frac{1}{4} \cdot (1 - \tanh^2(\chi)) \end{aligned} \quad (\text{C.5})$$

or that

$$4 \cdot (f^2(\chi) - f(\chi)) = (1 - \tanh^2(\chi)) \quad (\text{C.6})$$

Since the derivative of eqn. (C.2) is equal to

$$f'(\chi) = \frac{1}{2} (1 - \tanh^2(\chi)) \quad (\text{C.7})$$

we substitute eqn. (C.6) into eqn. (C.7) to get eqn. (C.3) as follows

$$\begin{aligned} f'(x) &= \frac{1}{2}(1 - \tanh^2(x)) = \frac{1}{2} \cdot 4 \cdot (f^2(x) - f(x)) = 2 \cdot (f^2(x) - f(x)) \\ &= 2f(x) \cdot (f(x) - 1) \end{aligned} \quad (\text{C.8})$$

## Appendix D. Proof of $\mathbf{P}^T \mathbf{P} = \mathbf{I}$

Let  $\mathbf{P}$  be a permutation matrix, where

$$\mathbf{P} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

is its singular value decomposition (SVD). Then the transpose of the permutation matrix is expressed as

$$\mathbf{P}^T = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T$$

Since  $\mathbf{P}$  is a square matrix,  $\mathbf{U}^T = \mathbf{U}^{-1}$ ,  $\mathbf{V}^T = \mathbf{V}^{-1}$  and these result from properties of the SVD. Also  $\mathbf{\Sigma}^T = \mathbf{\Sigma} = \mathbf{\Sigma}^{-1} = \mathbf{I}$  since any permutation matrix has all singular values equal to one. Given this,

$$\begin{aligned}\mathbf{P}^T &= \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T \\ &= \mathbf{V}^{-T}\mathbf{\Sigma}^{-1}\mathbf{U}^{-1} \\ &= (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^{-1} \\ &= \mathbf{P}^{-1}\end{aligned}$$

Multiplying both sides of this equality by  $\mathbf{P}$  on the right, yields the completed proof:

$$\mathbf{P}^T \mathbf{P} = \mathbf{I}.$$