8-2005

Inverse Problems in Image Processing: Blind Image Restoration

Ravi Viswanathan
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Ravi Viswanathan entitled "Inverse Problems in Image Processing: Blind Image Restoration." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

Mohamed R. Mahfouz, Major Professor

We have read this thesis and recommend its acceptance:

Anthony E. English, Aly Fathy

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Ravi Viswanathan entitled "Inverse Problems in Image Processing: Blind Image Restoration." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

Mohamed R. Mahfouz__
Major Professor

We have read this thesis
and recommend its acceptance:

Anthony E. English_____

Aly Fathy_____

Accepted for the Council:

__Anne Mayhew_____
Vice Chancellor and
Dean of Graduate Studies

(Original signatures are on file with official student records.)

INVERSE PROBLEMS IN IMAGE PROCESSING: BLIND IMAGE RESTORATION

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Ravi Viswanathan
August 2005

Acknowledgements

I am indebted to my energetic advisor Prof. Mohamed R. Mahfouz for constantly encouraging me in my research focus and making my UT experience truly enlightening. I am also grateful to my other thesis committee members Prof. Anthony E. English and Prof. Aly Fathy for stimulating discussions and valuable comments.

Thanks also go to UT faculty and students whose creative energy and ideas helped discover the creative self in me. Many thanks to my group of friends who have helped keep my spirits high.

Thanks mom, dad and Jai. You guys were my prime source of strength.

Abstract

Blind Image Restoration pertains to the estimation of degradation in an image, without any prior knowledge of the degradation system, and using this estimation to help restore the original image. Original Image, in this case, refers to that version of the image before it experienced degradation. In this thesis, after estimating the degradation system in the form of Gaussian blur and noise, we employ Deconvolution to help restore the original image.

In this thesis, we use a Redundant Wavelet based technique to estimate blur in the image using high-frequency information in the image itself. Lipschitz exponent – a measure of local regularity of signals, is computed using the evolution of wavelet coefficients of singularities across scales. It has been shown before that this exponent is related to the blur in the image and we use it in this case to estimate the standard deviation of the Gaussian blur. The properties of wavelets enable us to compute the noise variance in the image. In this thesis, we employ two cases of deconvolution – A strictly Fourier domain Regularized Iterative Wiener filtering approach and A Fourier-Wavelet Cascaded approach with Regularized Iterative Wiener filtering - to compute an estimate of the image to be restored using the blur and noise variance information that was earlier computed.

The estimated value of standard deviation of the blur helped obtain robust estimates with deconvolution. It can be observed from the results that Fourier domain Regularized Iterative Wiener filtering provides a more stable output estimate than the Iterative Filtering with Additive Correction methods, especially when the number of iterations employed is more. The Fourier-Wavelet Cascaded deconvolution seems to be

image dependent with regards to performance although it outperforms the strictly Fourier domain deconvolution approach in some cases, as can be gauged from the visual quality and Mean Squared Error.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Blind Image Restoration

Any image that is the output of a sensor is the result of the convolution of the analog input data to the sensor along with the Point Spread Function (PSF) of the sensor. This PSF is commonly termed 'blur'. This undesirable phenomenon of Blur occurs in a wide range of sensors – from Astronomy to Medical Imaging to Microscopy. In a lot of cases, valuable image information is lost due to the presence of blur. In order to restore this blurred image, the 'blur' would have to be estimated so that the original image can be restored to the best extent possible. This blur on the resulting image depends not only on the optics of the sensor but also on the distance between the sensor and the specimen that is being imaged. In order to first estimate blur, we would need to focus on parts of the image that would be affected most by it. Previous approaches in this field formulated a method for blur estimation using derivatives of the Gaussian Point Spread Function (PSF) [20, 21] and Steerable Pyramids [22] to determine the variance of the Gaussian blur.

In this thesis, we use redundant wavelet-based decomposition to break down the image into various scales. Blurring of low frequency components does not change the information content as much as blurring of high frequency information. A wavelet analysis of edge information in the image is implemented, to compute the center of gravity of Lipschitz exponents – a metric to compute local regularity of signals, corresponding to edges in the image. This Center of Gravity is then used to compute the blur. This process is first implemented for a set of training images, whose blur

information is already known, to derive certain proportionality constants to relate the Lipschitz exponent with the blur. These proportionality constants are then applied to compute the blur for the test set images. The variance of the additive noise present in many an input image is also estimated in this thesis, utilizing properties of the wavelet transform.

Given an observation that is comprised of an input image first degraded by linear time-invariant (LTI) convolution with a known impulse response and then corrupted by additive noise, deconvolution aims to estimate the input image. Deconvolution is extremely important in applications such as satellite imaging and seismic imaging. One of the earliest works in this regard would be Wiener filtering, which was improved upon by the Iterative Wiener filtering introduced by A.D.Hillery and R.T.Chin [9]. Neelsh et al improved the state of the art by performing Wiener filtering in the Fourier domain followed by Wavelet domain Wiener filtering [11].

In this thesis, we implement a method consisting of Iterative Wiener filtering with additive correction in the Fourier domain followed by Wavelet domain Wiener filtering. This method utilizes the blur and noise information that we computed in order to restore the original image. We also implement the already existing methods in order to compare the results.

The value of PSF estimated for the training images using our algorithm was within 10% of the actual value. The PSF estimation for the test set images was also satisfactory as they produced robust estimates after deconvolution. It can be observed from the results that performance in the Fourier domain gets better with higher number of iterations. It should be added that the Iterative Wiener filtering with Additive Correction approach provides unstable results, even in the Fourier domain, as in some cases, the Mean Squared Error and the visual quality deteriorate with higher number of iterations, and improve qualitatively in other cases.

The performance of the Fourier-Wavelet Cascaded deconvolution appears dependent on the provided image and seems relatively unstable when compared to the former approach, although the results in this method are far superior in some cases, especially in the test set images, in our case.

1.2 Image Deconvolution – Biomedical Applications

Our general understanding of Biological Sciences has been enhanced to a good extent by Microscopy techniques in general and Light Microscopy in particular. Of late, rapid advances in imaging and image processing algorithms along with staining techniques have helped unveil a variety of events in living cells in real time. High-resolution Imaging of biological specimen for biological and bio-medical applications has gained importance because of the inherent clarity of the image that it possesses.

In images obtained using Light Microscopy, besides the in-focus structures, the images usually also contain out-of-focus light from other parts of the object, causing haze and severe axial blur. This is even the case for precise equipment such as a Confocal Laser Scanning Microscope, where an attempt is made to remove most of the out-of-focus light from the image by a pinhole system. In this thesis, examples have been used to illustrate the effectiveness of the Image restoration algorithm in restoring images obtained from confocal microscopy.

Chapter 2

PSF Estimation

2.1 Blur – A Mathematical Representation

The formation of an image from a sensor can be treated as a classical discrete-time convolution process. As can be seen in Figure 2.1, the observed image consists of an unknown image *f* that is degraded first by circular convolution with an impulse response from a linear time-invariant (LTI) system *H* and then corrupted by additive Gaussian noise γ .

The impulse response of the LTI system is called the Point Spread Function of the system, or commonly termed 'blur'. In this thesis, we assume that the Gaussian noise associated with this PSF degradation is separate from the additive Gaussian noise γ that is shown in Figure 2.1.

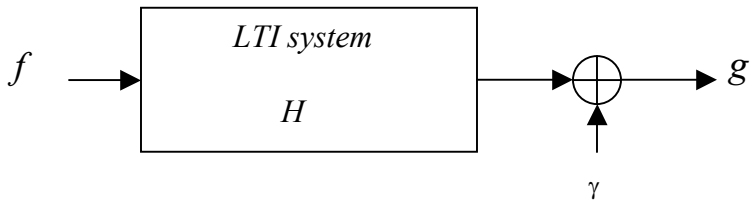


Figure 2.1: Degradation of an image by LTI system followed by additive noise.

Image blur can be modeled thus.

$$g(x,y) = (h * f)(x,y) \tag{2.1}$$

g being the blurred image, f the unknown sharp image and h the point spread function (PSF). The symbol * represents the convolution operator, which models the image blur. It is in fact the response of the imaging system to an ideal point source.

In order to first estimate blur, we would need to focus on parts of the image that would be affected most by it. Any information in a signal can be classified into Low frequency and High frequency components. Blurring of low frequency components does not change the information content as much as blurring of high frequency information. As a result, a wavelet analysis of edge information in the image would help compute the center of gravity of high frequency information. This Center of Gravity is then used to compute the blur.

2.2 Principle for Blur Estimation

In this thesis, we are basing our PSF estimation on the sharpness of edge information in the image across different scales. For this, a wavelet transform of the image is computed and the Lipschitz exponent in all points where a change in intensity is found is calculated. Lipschitz exponent is an indicator of the smoothness of the image at a certain point. Mathematically stating, it is a measure of how many times the image is differentiable in a certain point. Lipschitz exponent values typically vary from –1 for Dirac impulses, 0 for step responses. It tends towards unity as low frequency content increases. Upon wavelet transformation, Lipschitz exponent at desired points in the image can be evaluated by observing the wavelet coefficients for those spatial positions across scales. For intensity changes that are sharper, wavelet coefficients reduce in magnitude while they increase for low frequency information as we move from high-resolution scales to lower resolution scales. Figure 2.2 shows the evolution of wavelet coefficients across scales.

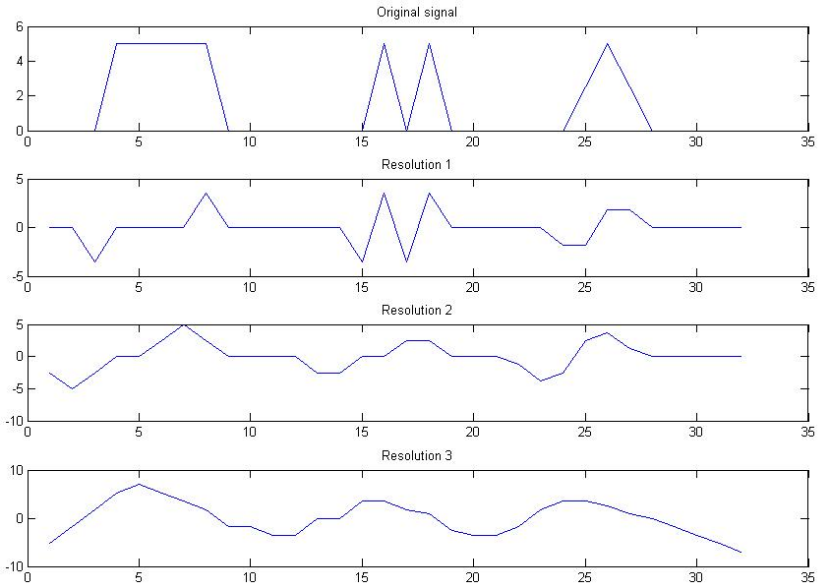


Figure 2.2: Evolution of wavelet coefficients across scales

Mallat has shown in [1] that the lipschitz exponent can be computed from this rate of change of wavelet coefficients across scales. This is the fundamental theory on which PSF Estimation is based in this work. Using Least Squares exponential fitting, the Lipschitz coefficient for each wavelet trace that provides the best fit is found. The mean for this set is related to the blur. Note that in all these computations, only the sharpest edges are used and thus the Lipschitz exponents are basically limited to those transitions with large amplitude.

In this thesis, we consider a set of clean images, whose blur is negligible and degrade these with Gaussian blur of varying standard deviations. These degraded images form the training set for our work. Applying the blur estimation algorithm on these training set images provides us with proportionality constants that relate the Lipschitz exponent with the standard deviation of the blur in the image. We then apply these proportionality constants when computing the blur for the test set images, after computing the mean Lipschitz exponent for the test set images. A block diagram summarizing the experimental flow can be found in Figure 2.3.

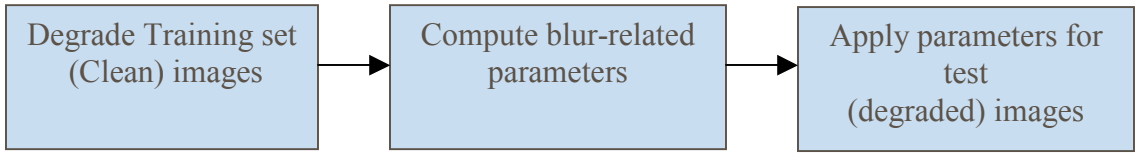
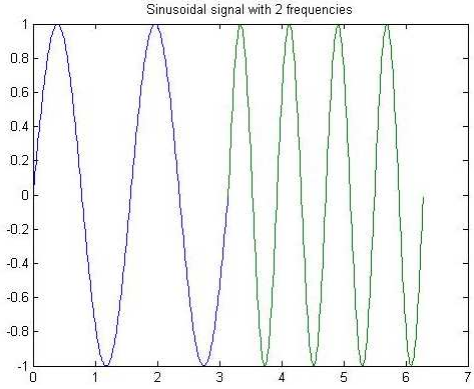


Figure 2.3 Experimental flow

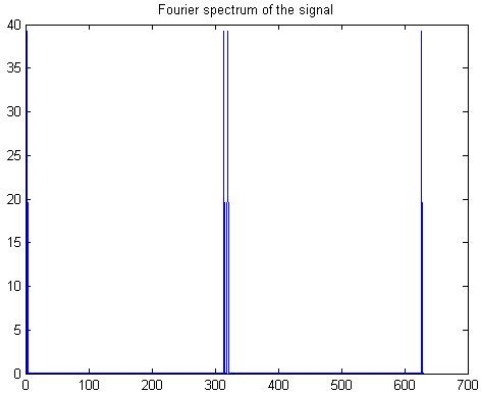
2. 3 The Wavelet Transform

Wavelets are a class of functions that are used for representing signals or other functions, not unlike the Fourier Transform, which consists of superposing sinusoids and co-sinusoids to represent other functions. The advantage of using a Wavelet transform is that provides time-frequency localization, as against the Fourier transform which provides only the frequency information of a signal. (Refer Figure 2.4)

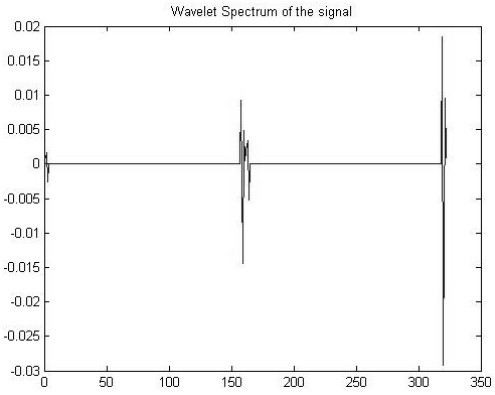
Consider a sinusoidal signal with 2 different non-zero frequencies at different time periods, as shown in Figure 2.4(a). A Fourier transform of this signal would yield 2 peaks, corresponding to the two frequency components in the signal. It does not provide information of the time at which the frequency occurs. However, a wavelet transform of such a signal provides a time-frequency representation, in that it provides the frequency information for every point in time. The difference between a wavelet representation and a Fourier representation is the resolution or scale, a critical parameter in Wavelet analysis. Wavelet based algorithms are employed on data at different scales, with the observed features being different as we go across scales. The observed features, in the simplest Mathematical case would be the values upon Wavelet transformation. We obtain general characteristics of the object of interest when we look at the signal representing it in a small window i.e. at a coarser scale. At fine scales, we observe many more particular characteristics of the object under study. Like sinusoids, which are the basis functions in Fourier analysis, a variety of ‘mother’ wavelets are available for wavelet analysis, among which Daubechies [3] is one of the most popular.



(a) Sinusoidal signal with 2 non-zero frequency components



(b) Fourier spectrum of the signal



(c) Wavelet spectrum of the signal.

Figure 2.4: Signal analysis in different domains.

To overcome the limitations of Fourier analysis such as resolution, the continuous wavelet transform was developed. The continuous wavelet transform is defined as follows

$$CWT_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^* \left(\frac{t - \tau}{s} \right) dt \tag{2.2}$$

As can be observed from the above equation, the signal is a function of time τ and the resolution or scale denoted by s. Note that the mother wavelet, which in our case is Daubechies is denoted by ψ.

2.3.1 Why Wavelets

In most real-world problems, high frequency components occur only in short spurts while low frequency components normally last for the entire duration of the signal. The purpose of using the scaling operator in wavelets is to dilate or compress the signal under study. At higher scales, we observe general features of the signal while lower or finer scales correspond to compressed signals and help observe the specifics of signal at various time points.

As can be seen from the CWT equation, let $x(t)$ be the signal under study. As was already discussed, Daubechies is used as the mother wavelet so as to serve as the basis function for all windows in this wavelet analysis. Signal analysis is done under windows, which are scaled versions of the fundamental mother wavelet. Given this set-up, the CWT is computed for a scale value of 1 and then on for larger values i.e. observes the highest frequency components of the signal first and gradually moves on towards lower frequencies. Almost all real-world signals are band-limited and hence require analysis only a limited number of scales to get a good understanding of the signal.

An understanding of the time-frequency representation can be underlined thus. The mother wavelet is used to create a window at the time origin of the signal. The product of the signal and the wavelet at the first scale is then computed and is integrated over all time and is then multiplied by the constant detailed in the equation so as to analyze the transformed signal under uniform energy levels across all scales. Thus we get the value of the transform for time t=0 and for s=1. We progressively move on towards increasing time for the same scale and compute the wavelet-transformed value at each point. Once all time points are covered, we repeat the process for higher values of scale. Because we use time-shifting as well as different scale values, the signal achieves time and frequency localization.

It is understandable that under such a process, if the signal has frequency components that correspond to the window used (scale), a large value would result. Low frequency components result at higher scales and hence there is not much difference in wavelet values, even at time locations where there was a significant difference at lower scales. In summary, it can be said that the Wavelet transform provides excellent time and poor frequency resolution at lower scales and poor time and good frequency resolution at higher scales.

2.3.2 The Discrete Wavelet Transform

When dealing with discrete data as in digital signals and images, it becomes necessary to formulate a discrete formulation of the Continuous Wavelet Transform that was detailed earlier. A discretized form of the CWT explained earlier is not the solution for this problem, given that the wavelet outputs are themselves a sampled version of the CWT and the information it provides is redundant. Processing redundancy takes away a significant portion of computational time and resources. Hence, a Discrete Wavelet Transform attempts to do away with redundancy reduce computational time and effort while at the same time retaining sufficient information for efficient signal reconstruction.

Not unlike the sampling employed in the Discrete Fourier Transform method, sampling in the time-frequency domain is applied in DWT. However, instead of uniform sampling, the sampling in this case can be varied across scales to save computational time and resources. This can be achieved because progressively smaller values of samples would be necessary as we move from lower scales to higher scales for having a good signal representation.

The *1-D* discrete wavelet transform (DWT) represents a 1-D continuous-time signal $x(t) \in L^2([0,1))$, $t \in [0,1)$, in terms of shifted versions of a low-pass scaling function ϕ and shifted and dilated versions of a prototype band-pass wavelet function ψ. For special choices of ϕ and ψ, the functions $\psi_{j,l}(t) = 2^{j/2}\psi(2^j t-l)$ and $\phi_{j,l}(t) = 2^{j/2}\phi(2^j t-l)$ with $j,l \in Z$ form an orthonormal basis. The j parameter corresponds to the *scale* of the analysis, while the l parameter corresponds to the *location*.

A finite-resolution approximation x^J to x is given by

$$x^J(t) = \sum_{l=0}^{N_{j0}-1} S_{j_0,l}\phi_{j_0,l}(t) + \sum_{j=j_0}^{J}\sum_{l=0}^{N_j-1} w_{j,l}\psi_{j,l}(t) \qquad (2.3)$$

with the scaling coefficients $s_{j0,l} = \langle x, \phi_{j0,l} \rangle$ and wavelet coefficients $w_{j,l} = \langle x, \psi_{j,l} \rangle$

The parameter J controls the resolution of the wavelet reconstruction x^J of x. In fact, the L_2 norm $\|x^J - x\|_2 \to 0$ as $J \to \infty$.

Applying a DWT on the image shown in Figure 2.5(a) results in a multi-resolution image shown in Figure 2.5(b). The image in the last quadrant is that of the highest resolution image and contains 'detail' coefficients. The top-right and bottom-left quadrants contain the wavelet transform of the original image in horizontal and vertical orientations at the highest resolution. The top-left quadrant contains images corresponding to wavelet transform of the original image at successively coarser scales.

(a) Original image (b) Upon DWT

Figure 2.5: Discrete Wavelet Transform on an image

2.3.4 Economy of Wavelet Representations

In smoothness spaces such as Besov's, Wavelets helps provide economical representation of the signals under study [4]. A Besov space $B^s_{p,q}$ contains functions with s derivatives in L_p with q measuring finer smoothness distinctions. Because the signals we consider are images, Besov spaces are significant as they encompass images with singularities.

2.4 Mallat's Relationships – The Lipschitz Exponent

Edge information in an image is often the most important, mainly because these singularities define the boundary of the object contained in the image. Fourier Transform has been used as a tool for studying singularities for a long time but it serves the purpose of a general study of the signal and does not help find location and spatial distributions of edge information. On the other hand, Wavelets emerged as a tool that helped characterize the local regularity of signals by decomposing the signal into parts that are localized in both space and frequency. Lipschitz exponents are a particular class of Mathematical functions that help provide a measure of this local regularity of signals [5]. The relationship between Lipschitz exponents and their use in deriving local regularity statistics of a signal has been established [1,2].

A function *f(x)* is said to satisfy the Lipschitz condition of order α at *x=0* if

$$|f(h) - f(0)| \leq B|h|^{\beta} \tag{2.5}$$

for all $|h| < \in$, where *B* and β are independent of *h*, $\beta > 0$ and α is an upper bound for all β for which a finite *B* exists.

2.4.1 Definition I

- Let n be a positive integer and $n \leq \alpha \leq n+1$. A function *f(x)* is said to be Lipschitz α, at x_0, if and only if there exists two constant *A* and $h_0 > 0$ and a polynomial of order *n*, $P_n(x)$ such that for $h < h_0$

$$|f(x_0 + h) - P_n(h)| \leq A|h|^{\alpha} \tag{2.6}$$

- The function *f(x)* is uniformly Lipschitz α over the open interval]*a, b*[, if and only if there exists a constant *A* and for any $x_0 \in$]*a, b*[, there exists a polynomial of order *n*, $P_n(h)$ such that equation (1) is satisfied if $x_0 + h \in$]*a ,b*[
- We call Lipschitz regularity of *f(x)* and x_0, the superior bound of all values α such that *f(x)* is Lipschitz alpha at x_0.
- We say that a function is singular at x_0, if it is not Lipschitz 1 and x_0.

Lipschitz exponent at very smooth points in a function are nearly 1 as they are continuously differentiable at that point. Given that the derivative of the function is bounded but discontinuous at x_0, and following definition 1, *f(x)* is not singular at x_0. It can be concluded that if the Lipschitz regularity α_0 of *f(x)* satisfies $n \leq \alpha \leq n+1$, then we know that *f(x)* is *n* times differentiable at x_0, but its n^{th} derivative is singular at x_0 and α_0 characterizes this singularity [1,2].

2.4.2 Definition II

Let *f*(x) be a tempered distribution of finite order. Let α be a non-integer real number and [*a, b*] an interval of R. The distribution *f(x)* is said to be uniformly Lipschitz α on] *a, b*[if and only if its primitive is uniformly Lipschitz α+*1* on] *a, b*[.

The second order primitive of a Dirac is a function which is piecewise linear in the neighborhood *x=0*. This function is uniformly *Lipschitz 1* in the neighborhood of 0 and thus uniformly Lipschitz α for α<*1*. As a consequence of definition 2, a Dirac is uniformly Lipschitz α, for α<-*1*, in the neighborhood of 0. Definition 2 has a more general approach as uniform Lipschitz exponents are defined over intervals but not at points. It is possible to locate isolated singularities in function *f(x)* thus. If *f(x)* is uniformly Lipschitz *a* over an interval [*a, b*] with $x_0 \in$]*a ,b*[, and *f(x)* is uniformly *Lipschitz 1* over any subinterval of]*a, b*[that does not include x_0. As an example, a Dirac centered at 0 has an isolated singularity at *x=0* whose Lipschitz regularity is –1.

It is proved in [1] that given $f(x) \in L^2(R)$ and [*a, b*] an interval of *R* and if *0*<α<*1*, then for any *e>0*, *f(x)* is uniformly Lipschitz α over] *a+e, b-e*[if and only if for any *e>0*, there exists a constant A, such that for x ∈] *a+e, b-e*[and *s*>0.

$$|Wf(s,x)| \le As^{\alpha} \tag{2.7}$$

If $f(x) \in L^2(R)$, for any scale s_0>0, by applying the Cauchy-Schwartz inequality, it can be proved that the function |*Wf*(s,x)| is bounded over the domain $s>s_0$. In order to extend the theorem detailed above to exponents larger than 1, it is necessary that the wavelet has sufficient number of vanishing moments. A wavelet *W*(x) is said to have '*n*' vanishing moments, if and only if for all positive integer *k*<*n*, it satisfies

$$\int_{-\infty}^{+\infty} x^k \psi(x)\,dx = 0$$

The theorem detailed by Mallat remains valid as long as $0<a<n$ where n is the number of vanishing moments of the wavelet.

2.4.3 Vanishing Moments

Supposing 'n' denotes the number of vanishing moments of the wavelet used, then $W'(w)$ is n times continuously differentiable and we can infer from the theorem that $W'(w)$ has a zero of order n in w=0. Hence, for any integer value $p<n$, $W'(w)$ can thus be factorized into

$$W'(w) = (iw)^p W'(w)$$

This can be translated in the spatial domain thus

$$\psi(x) = \frac{d^p \psi^1(x)}{d^p x} \tag{2.8}$$

The wavelet transform of $f(x)$ with respect to the wavelet $W(x)$ is in effect, equal to the wavelet transform of its p^{th} derivative, computed with the wavelet $W'(x)$, and multiplied by s^p . Let p be any integer such that $0<\alpha-p<1$. The function $f(x)$ is uniformly Lipschitz α on an interval] a, b[if and only if $(d^p f)/(dx^p)$ is uniformly Lipschitz α-p on the same interval.

2.5 Singularity Detection

One of the earliest approaches to detect edges using wavelets was done by Grossman [6], using a Hardy function. A Hardy function is a complex function whose Fourier Transform equals zero for frequency values below zero. Smoothing functions such as Gaussians are employed to illustrate the singularity detection in [1]. If $t(x)$ be the smoothing function, then $t_s(x) = 1/s.\ t(x/s)$. It is clear that given such a smoothing function as a basis, the singularities at any scale s can be located from the sharp variation points of

f(x) smoothed by $t_s(x)$. *W'(x)* and *W''(x)*, the two wavelets are the first and second differential of *t(x)* with respect to *x*. Consequently, the two wavelet transforms *W'f*(*s,x*) and *W''f*(*s,x*) are proportional to *W'*(*x*) and *W''*(*x*) smoothed by $t_s(x)$. It can be inferred from this relationship that given any scale *s*, the local extreme of *W'f*(*s,x*) along the *x* variable, correspond to the zero-crossings of *W''f*(*s,x*) and to the inflection points of $f^{*}t_s(x)$. It is possible to define a particular Hardy wavelet such that the phase of the wavelet transform remains constant or changes sign, along the set of smooth curves that result in the zero-crossings of *W''f*(*s,x*). Either an absolute maximum or minimum results at a point of a function whose primitive would correspond to an inflection point. As our purpose in employing wavelets mainly involves singularity detection, we focus on only the local maxima values of |*W'f*(*s,x*)|.

2.6 Wavelet Transform Modulus Maxima

If *Wf*(*s,x*) be the wavelet transform of a function *f*(*x*), then the point (s_0,x_0) is called 'Modulus Maxima' if $|Wf(s_0,x)| \le |Wf(s_0,x_0)|$, where *x* belongs to the neighborhood of x_0. A maxima line is one, which contains a trace of modulus maxima values. A function is not singular in any neighborhood where its wavelet transform has no modulus maxima at fine scales [1, 2]. It is clear from such a theoretical formulation that all the edges in an image function *f* can be located by following the maxima lines when the scale goes to zero.

2.7 Modulus Maxima and Lipschitz Exponent – The Relationship

If *f(x)* is the function whose wavelet transform is well defined over]*a, b*[and if $x_0 \in]a,b[$ and there exists a scale s_0>0 and a constant *K*, such that for $x \in]a,b[$ and $s<s_0$, all of the Wavelet modulus maxima values of this function belong to a domain defined by

$$\left|x - x_0\right| \leq Cs \tag{2.9}$$

Then, at all points $x_1 \in]a,b[$, $x_1 \neq x_0$, $f(x)$ is uniformly Lipschitz α in a neighborhood of x_1. The function $f(x)$ is Lipschitz α at x_0, if and only if there exists a constant A such that at each modulus maxima (s,x) in the domain specified by (2.9)

$$\left|Wf(s,x)\right| \leq As^{\alpha} \tag{2.10}$$

By computing the coefficient α in the above equation such that $A\text{s}^{\alpha}$ approximates the decay of the wavelet coefficients of a point (s, x) over different scales. Hence, we obtain the Lipschitz exponents that can help characterize the singularities in a signal or edges, in case of images. Step changes in signal correspond to Lipschitz exponents of 0 while sharp irregularities or Dirac's correspond to negative values of Lipschitz exponents.

2.8 Computation of Lipschitz Exponent

For a function of the form

$$y = A(2)^{Bx} \tag{2.11}$$

Taking logarithm to the base 2 on both sides,

$$log\,2(y) = log\,2(A) + Bx \tag{2.12}$$

A typical plot obtained using Least Squares fitting for a case such as the one detailed in (2.11) can be found in Figure 2.6. In practice, we use the Least Squares fitting algorithm to obtain a curve fit for this case. The fundamental basis of a Least Squares approach is to minimize the sum of squares of offsets of points from the curve [5].

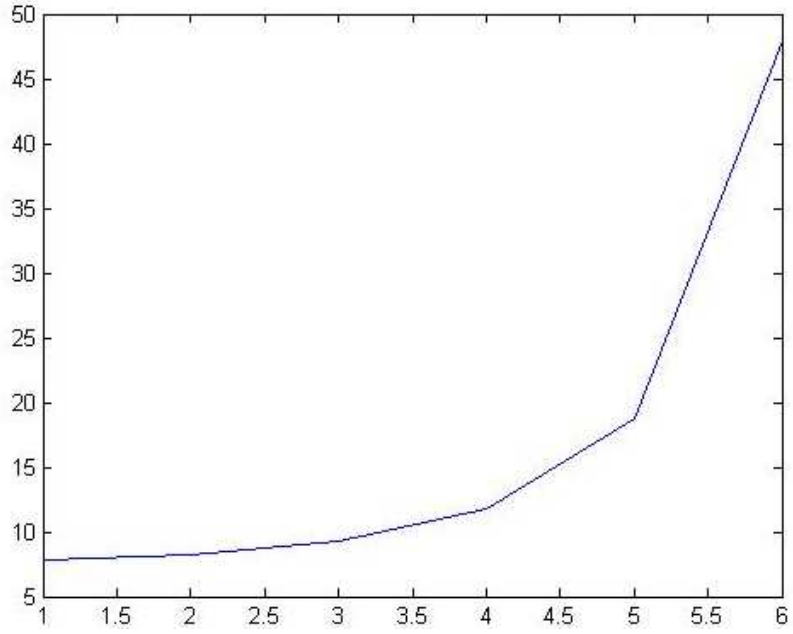


Figure 2.6: Fitting at dyadic powers

Using the Least Squares fitting for such a case provides the best-fit values as follows

$$a = \frac{\sum_{i=1}^{n} log\,2(y_i) \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i\ log\,2(y_i)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$b = \frac{n\sum_{i=1}^{n} x_i\, log\,2(y_i) - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} log\,2(y_i)}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \qquad (2.13)$$

The value of B in (2.12), as given by this fit would be equal to b obtained in (2.13). The value of A in (2.12) would be equal to 2^a, given the value of a obtained in (2.13).

Our interest in this fitting stems from the Lipshitz relationship with Wavelet modulus maxima, which has the same form as the above.

$$\left|Wf(s,x)\right| \le As^{\alpha} \tag{2.14}$$

where s, being the scale, has the form of $(2)^j$

Rewriting (2.14) in logarithmic terms,

$$log\,2(\left|Wf(s,x)\right|) = log\,2(A) + \alpha j \tag{2.15}$$

Applying the fitting detailed in (2.13) to the relationship in (2.15), we get the value of the Lipschitz exponent α.

But the fit described gives greater weights to smaller values of y. Better-fit values can be obtained by minimizing the function

$$\sum_{i=1}^{n} y_i\,(log\,2(y_i) - a - bx_i)^2 \tag{2.16}$$

This can be solved using Least Squares fitting.

$$a\sum_{i=1}^{n} y_i + b\sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} y_i\,log\,2(y_i)$$

$$a\sum_{i=1}^{n} x_i y_i + b\sum_{i=1}^{n} x_i^2 y_i = \sum_{i=1}^{n} x_i y_i\,log\,2(y_i)$$

$$\begin{bmatrix} \sum_{i=1}^{n} y_i & \sum_{i=1}^{n} x_i y_i \\ \sum_{i=1}^{n} x_i y_i & \sum_{i=1}^{n} x_i^2 y_i \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i\,log\,2(y_i) \\ \sum_{i=1}^{n} x_i y_i\,log\,2(y_i) \end{bmatrix} \tag{2.17}$$

$$a = \frac{\sum_{i=1}^{n} y_i \log 2(y_i) \sum_{i=1}^{n} x_i^2 y_i - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} x_i y_i \log 2(y_i)}{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 y_i - \left(\sum_{i=1}^{n} x_i y_i \right)^2}$$

$$b = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i y_i \log 2(y_i) - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} y_i \log 2(y_i)}{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 y_i - \left(\sum_{i=1}^{n} x_i y_i \right)^2} \tag{2.18}$$

Applying this fit in (2.18) helps obtain better-fit values for the Lipschitz exponent. In this thesis, because we assume that the noise associated with the estimates is negligible, we employ such a trivial least squares fitting approach.

2.9 Blur and Lipschitz Exponent – The Relationship

The smoothing functions used in [1,2] to illustrate the use of Lipschitz exponent for singularity detection were Gaussian. In most blind deconvolution problems, assuming a Gaussian Point Spread Function works well, as it is a good generalization of the problem. A Gaussian function is of the form

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} exp\left(\frac{-(x-\bar{x})^2}{2\sigma^2} \right) \tag{2.19}$$

where $\bar{x}$ is the mean of the distribution.

In this thesis, we assume a Gaussian blur. Hence, the most critical parameter to be found is the standard deviation. As can be deduced from (2.19), the relationship between the standard deviation and mean can be expressed thus,

$$\sigma \approx exp\left(\frac{-(x-\bar{x})^2}{2\sigma^2}\right) \tag{2.20}$$

In [1], the relationship between lipschitz exponent and the Wavelet modulus maxima are derived. The smoothing functions in this case, whose derivatives constitute the Wavelet basis functions, are usually Gaussians in edge detection and other computer vision problems. It can be summarized that the wavelet evolution across scales thus depends on the following factors [2].

1. The regularity of the original underlying signal.
2. The properties of the wavelet basis functions used.
3. The blur of the signal at the given position.

In our case, computation of the lipschitz exponent helps us know the regularity of the signal (*factor 1*). And we know the property of the wavelet basis function used, which is Daubechies (*factor 2*). With thus, we can calculate blur as we have knowledge of 2 out of 3 factors. In this thesis, we experiment with various exponential and other non-linear fitting algorithms in order to derive a relationship between the standard deviation of the blur and the mean value of lipschitz exponent [19].

If there are n_k wavelet modulus maxima traces for a lipschitz exponent α_k, then the mean value is given by

$$\text{Mean} = \frac{\sum_k n_k \alpha_k}{\sum_k n_k} \tag{2.21}$$

In effect, the relationship between the mean lipschitz exponent from (2.21) and the standard deviation of the blur can be expressed thus

$$\sigma = a\, exp(bx) \tag{2.22}$$

x in this case, is the mean value of the lipschitz exponent.

2.10 Computation of Constants

In order to compute *a* and *b* described in the above blur relationship (5), we apply our algorithm on a variety of images taken from the system whose output images we wish to deblur. These *training images* are ones whose blur values we already know. A histogram of Lipschitz exponents obtained for each training image is constructed and the center of gravity of this histogram is evaluated. Applying the standard deviation of the blur for images with known blur and the center of gravity of the lipschitz exponents for the corresponding image in (5), we solve for a and b using Least Squares fitting.

For a function of the form

$$y = A\, e^{Bx} \tag{2.23}$$

Taking logarithm to the base 2 on both sides gives

$$ln(y) = ln(A) + Bx \tag{2.24}$$

A typical curve obtained using Least Squares fitting for a case as in (2.23) can be seen in Figure 2.7.

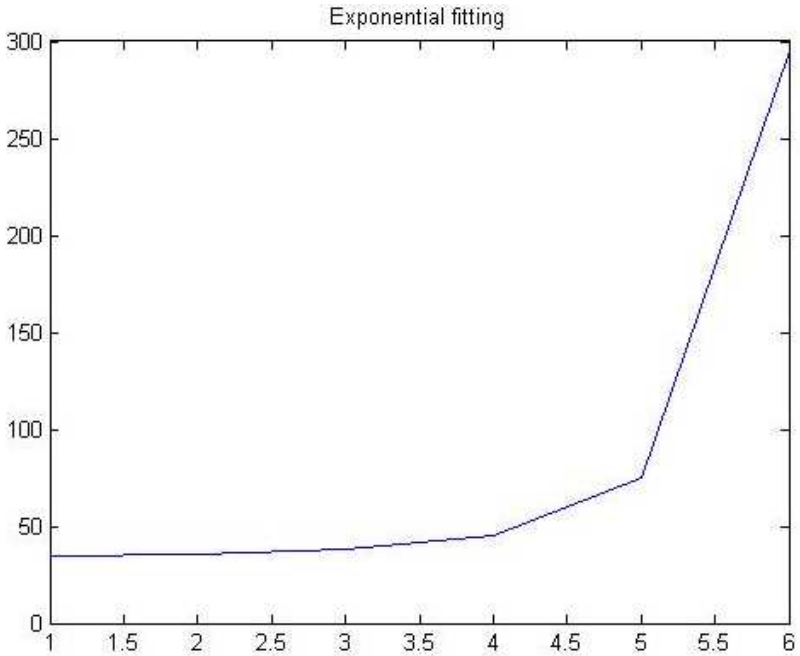


Figure 2.7: Exponential fitting curve

Using the Least Squares fitting for such a case provides the best-fit values as follows

$$a = \frac{\sum_{i=1}^{n} ln(y_i) \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i \, ln(y_i)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2}$$

$$b = \frac{n \sum_{i=1}^{n} x_i ln(y_i) - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} ln(y_i)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2} \qquad (2.25)$$

Better-fit values can be obtained by minimizing the function

$$\sum_{i=1}^{n} y_i \, (ln(y_i) - a - bx_i)^2 \qquad (2.26)$$

This can be solved using least squares fitting. Employing Least Squares fitting, we get

$$a = \frac{\sum_{i=1}^{n} y_i ln(y_i) \sum_{i=1}^{n} x_i^2 y_i - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} x_i \, y_i ln(y_i)}{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 y_i - \left(\sum_{i=1}^{n} x_i y_i \right)^2}$$

$$b = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i y_i ln(y_i) - \sum_{i=1}^{n} x_i y_i \sum_{i=1}^{n} y_i ln(y_i)}{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 y_i - \left(\sum_{i=1}^{n} x_i y_i \right)^2} \qquad (2.27)$$

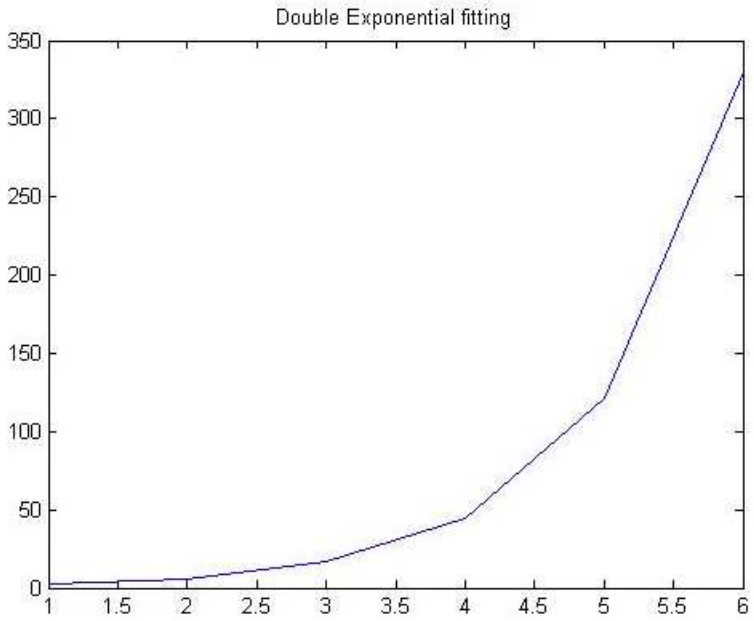


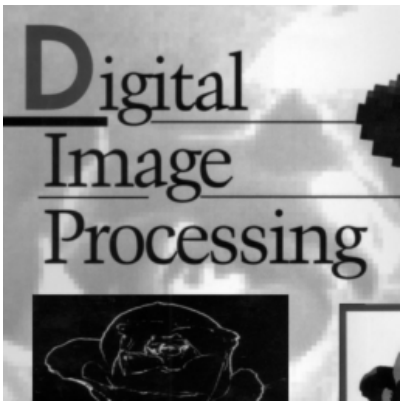
Figure 2.8: Double Exponential fitting curve

The exponential fitting scheme in (2.23) can be further improved upon by using a double exponential fitting

$$y = A\, e^{Bx} + C\, e^{Dx} \tag{2.28}$$

Figure 2.8 shows the fitting curve obtained when employing a double-exponential case instead of the one used in (2.23). The advantage in using such a scheme is that the generalization performance of such a fitting would be better. Hence, the blur estimation for the test images would be more robust.

Figure 2.9(a) is the original training set image, which is then degraded using a Gaussian blur of standard deviation 4.0 to form the image in Figure 2.9(b). Using our algorithm, we compute the mean Lipschitz exponent for this image and consequently the standard deviation of the Gaussian blur applied to this otherwise clean image. Figure 2.9(d) shows the deconvolved result obtained using our blur estimate. Figures 2.9(c) and 2.9(e) show the result obtained after deconvolution using other blur estimates.

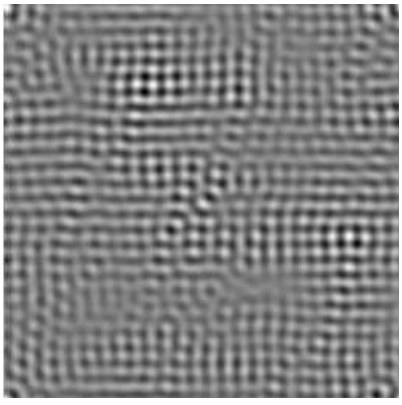


(a) Original training set image

(b) Degraded with Gaussian blur of SD (Standard Deviation) = 4.0

(c) Restored with SD=1.5

(d) Restored with SD=4.2



(e) Restored with SD=6.5

Figure 2.9: Image Restoration results for training image with different blur estimates.

After obtaining values for a and b, (2.22) is applied to evaluate the standard deviation of the Gaussian PSF for test images after computing the mean value of the Lipschitz exponent for that image. The computed value in (2.22) is an estimate of both the blur applied to the test image as well as the blur that already existed in the test image. We hence try to retain only that portion that corresponds to the already existing blur in the image and remove the applied blur portion.

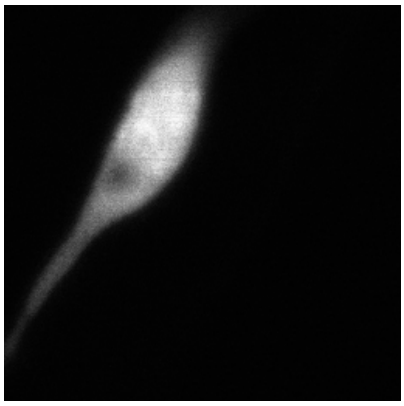
$$\sigma_{eff} = \sqrt{\sigma^2 - \sigma_{blEst}^2} \tag{2.29}$$

σ_{blEst}^2 is the estimated value of standard deviation σ_{bl}, found when employing the training set images.

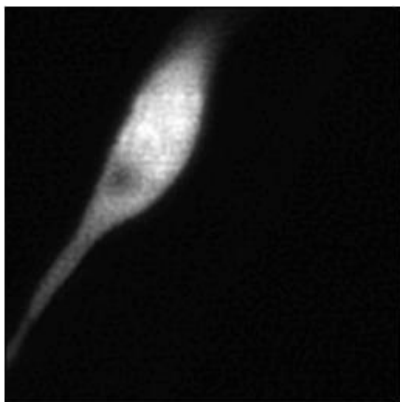
Figure 2.10(a) is the original test image, which is obtained from a Confocal Microscope. Using our algorithm, we compute the mean Lipschitz exponent for this image and consequently the standard deviation of the Gaussian blur already existing in this image. Figure 2.10(c) shows the deconvolved result obtained using our blur estimate. Figures 2.10(b) and 2.10(d) show the result obtained after deconvolution using other blur estimates.

2.11 Implementation Summary

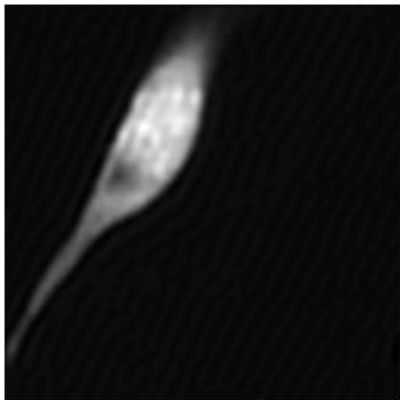
The wavelet transform of the image under consideration is computed, up to 3 resolution scales in our case. Hence, we have wavelet coefficient information for 3 dyadic scales – the detail information and the low frequency information. We compute a threshold value for the highest resolution band using the 95th percentile of the wavelet coefficients for the band. Because we wish to observe the variation of the wavelet coefficient corresponding to the sharpest edges across scales, such a value of threshold is chosen. This way, only the sharpest edge information is retained.



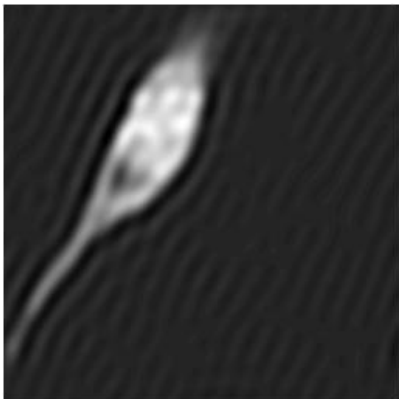
(a) Original test image (Courtesy: Ghent University)



(b) Restored with SD=1.5



(c) Restored with SD=2.8 (Our estimate)



(d) Restored with SD=4.5

Figure 2.10: Image Restoration results for test image with different blur estimates.

We compute the local maxima in every 4x4 neighborhood of the band and store it in an array. This array is then hard-thresholded using the thresh value stated earlier. The location of every non-zero value after the thresholding operation is noted. Using the location information, we look for the wavelet coefficients in the corresponding locations at the following dyadic scales. (Note that the location information is to be divided by 2 for the second scale and by 4 for the third scale). If there are n non-zero wavelet coefficients in the first scale after thresholding, now there needs to be n wavelet traces containing wavelet coefficients for the first, second and third scale. Hence, we effectively have an *n*x*3* matrix containing wavelet coefficients. Lipschitz exponent – a measure of singularity, and related to the wavelet modulus maxima, is computed using Least Squares fitting. The result is an array of length *n* containing lipschitz exponents for the sharpest edges in the image. A histogram is then constructed with these exponents and the mean is calculated. A set of training images whose blur is known, are used to find constants a and b in (2.22), after computing the mean of the lipschitz exponents corresponding to these images. After arriving at a value of a and b, using Least Squares fitting, we employ these constants in evaluating the standard deviation of the blur function for test images after computing the mean of the lipschitz exponent for these images.

Chapter 3

Fourier Domain Deconvolution

Deconvolution is the process of restoring the original image using the PSF and noise information. It is a fairly wide topic and finds application wherever blur is introduced due to the sensor. There is a wide range of algorithms for deconvolution and continue to grow with advances in Multi-resolution theory among others.

3.1 Problem Statement

In this work, we consider the discrete-time deconvolution problem. The observed samples *y(n)*, which is the sensor output, consists of the unknown desired signal samples *x(n)* degraded by an unknown impulse response *h* from a linear time-invariant (LTI) system *H* and corrupted by noise, which may be Gaussian or Poisson.

$$\begin{aligned} y(n) &:= Hx(n) + \gamma(n) \quad , n = 0, \ldots , N\text{-}1 \\ &:= (h \otimes x)(n) + \gamma(n) \end{aligned} \tag{3.1}$$

As stated earlier, we have knowledge of *y* and we try to estimate *h* using Lipschitz exponents. Further, we estimate noise variance in the image using wavelets. Using these known values, we seek to estimate *x*.

Operator inversion on *y* provides a trivial deconvolution estimate.

$$\tilde{x}(n) \cong H^{-1} y(n) = x(n) + H^{-1}\gamma(n) \tag{3.2}$$

But this estimate is poor on account of a variety of reasons, primarily because H is ill conditioned.

3.2 Computation of Noise Variance

In practical situations, the noise variance is usually unknown, as is the case in our work. In such cases, the standard deviation of noise can be estimated by using the median absolute value of the finest scale wavelet coefficients [7].

$$\sigma = \frac{median(W(1,x))}{0.6745} \tag{3.3}$$

$W(1,x)$ indicates the wavelet coefficients in the highest resolution band or the first scale.

In summary, the finest scale wavelet coefficients are ordered; the median of this array is computed and divided by a factor of 0.67 to obtain the standard deviation of noise for the image under study.

3.3 Linear Time Invariant Wiener Filtering

The Fourier domain is a popular one for estimating x from $\tilde{x}$. There are many filtering approaches in this domain among which LTI Wiener deconvolution filter is a robust choice. A significant advantage of employing deconvolution in the Fourier domain is that it represents the colored noise economically, as has been shown in [8]. Moreover, convolution simplifies to scalar Fourier operations in this domain. Therefore, (3.1) can be expressed in the Fourier domain thus,

$$Y(f_k) = H(f_k)X(f_k) + \Gamma(f_k) \tag{3.4}$$

Y, H, X and *Γ* being the respective Discrete Fourier Transforms (DFT) of *y, h, x and η.* Applying inversion in this domain, we get

$$\begin{aligned} &\tilde{X}(f_k) = X(f_k) + \frac{\Gamma(f_k)}{H(f_k)} \, if \left|H(f_k)\right| > 0, \\ &\tilde{X}(f_k) = 0, otherwise \end{aligned} \tag{3.5}$$

$\tilde{X}$ is the *DFT* of $\tilde{x}$. As is obvious from the equation itself, noise components where $\left|H(f_k)\right| \approx 0$ are particularly amplified during operator inversion.

Regularized Deconvolution is an efficient means of attenuating the amplified noise. The shrinkage provided by such an operation can be expressed thus

$$\lambda_k^f = \frac{\left|H(f_k)\right|^2}{\left|H(f_k)\right|^2 + \Lambda(f_k)} \tag{3.6}$$

$\Lambda(f_k) \geq 0$ is commonly referred to as the regularization term and controls the extent of shrinkage. The estimate of such an operation is given by

$$\tilde{X}_{\lambda^f}(f_k) = \tilde{X}(f_k)\lambda_k^f$$

$$= X(f_k)\left(\frac{\left|H(f_k)\right|^2}{\left|H(f_k)\right|^2 + \Lambda(f_k)}\right) + \frac{\Gamma(f_k)}{H(f_k)}\left(\frac{\left|H(f_k)\right|^2}{\left|H(f_k)\right|^2 + \Lambda(f_k)}\right)$$

$$= X_{\lambda^f}(f_k) + \frac{\Gamma_{\lambda^f}(f_k)}{H(f_k)} \tag{3.7}$$

The first and second terms in (3.7) represent the respective *DFT's* of the retained signal and the leaked noise.

3.4 Regularized Deconvolution

Some types of Regularized Deconvolution include *LTI Wiener* deconvolution [9] and *Tikhonov*-regularized deconvolution [10] and primarily differ in their choice of shrinkage.

The former regularization method consists of shrinkage with the following form.

$$\lambda_k^f = \frac{|H(f_k)|^2}{|H(f_k)|^2 + \alpha \frac{N\sigma^2}{|X(f_k)|^2}} \tag{3.8}$$

with *regularization parameter* = 1 to shrink more at frequencies where the signal power $|X(f_k)|^2$ is small.

If the value of noise variance is large compared to the Power Spectral Density (PSD) of the signal, the denominator term in (3.8) becomes large, resulting in a small value of λ_k^f. This, in effect, implies higher shrinkage. If the PSD of the signal is larger compared to the noise variance, then a larger value of λ_k^f results. Therefore, the shrinkage applied is smaller and more signal components are retained. This shrinkage effect therefore, affects the intensity profile of the resulting image (i.e.) the dynamic range of the deconvolved estimate could be affected directly depending on the shrinkage.

Tikhonov-regularized deconvolution, which is similar to LTI Wiener deconvolution assuming a flat signal spectrum $|X(f_k)|^2$, sets

$$\lambda_k^f = \frac{|H(f_k)|^2}{|H(f_k)|^2 + \tau} \quad , \tau > 0 \tag{3.9}$$

3.4.1 Choice of Shrinkage Term

In practice, an array of logarithmically spaced set of values with a median value of 1 is chosen as the initial set of values for α. This is so done because the difference between the various cases employing different shrinkage terms would be possibly much more if the values are logarithmically spaced instead of a linear displacement.

3.5 Fourier Domain Filtering

The degradation process, as can be seen in Figure 3.1 can be modeled as a degradation function, that operates on an input image and adds noise to this result to produce a degraded image. The objective of image restoration is to obtain an estimate of the original image using information gathered from the degraded image itself, the degradation function and the noise that corrupts it.

For the estimate to be as close as possible to the original image, we require that the precision of the degradation function estimated and the noise are excellent. In our case, we would first need to estimate the degradation function and the noise using the degraded image and then utilize this information to restore the original image. This process is called 'blind deconvolution'.

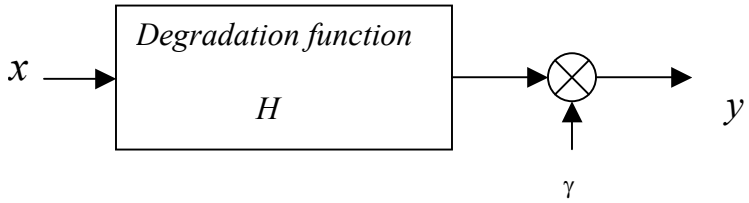


Figure 3.1: Image degradation process

3.5.1 Inverse Filtering

After obtaining an estimate of the degradation function H, we can attempt to restore the original image. The simplest inverse filtering would involve dividing the Fourier Transform of the degraded image Y by the Fourier Transform of the degradation function H to obtain an estimate.

$$\widetilde{X}(u,v) = \frac{Y(u,v)}{H(u,v)} \tag{3.10}$$

3.5.2 Wiener Filtering

The Inverse filtering mechanism is a very rudimentary one and does not handle noise effectively. Wiener filtering is a process that takes into account both the degradation function and the noise characteristics optimally during the restoration process. The goal in Wiener Filtering is to minimize the Mean Square Error between the restored image and the original image. The error measure can be written as

$$e = E\left\{(x - \tilde{x})^2\right\} \tag{3.11}$$

The fundamental assumptions in this case are that the noise and the image are uncorrelated and that the intensity profile in the estimate would be a linear function of that in the degraded image. Given this condition set, Wiener derived the following formulation to compute the minimum of the error function in the Fourier domain.

$$\widetilde{X}(u,v) = \left[\frac{H^*(u,v)P_{xx}(u,v)}{P_{xx}(u,v)|H(u,v)|^2 + P_{\eta\eta}(u,v)}\right]Y(u,v)$$

$$= \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \frac{P_{\eta\eta}(u,v)}{P_{xx}(u,v)}} \right] Y(u,v) \qquad (3.12)$$

$P_{\eta\eta}$ is the power spectrum of the noise; $P_{\eta\eta} = N\sigma^2$

N is the size of the signal. In our case, it would be the number of pixels in the image.

σ^2 is the noise variance.

P_{xx} is the power spectrum of the original image.

(3.12) summarizes the Wiener filter, which is hence also called the *minimum mean square error filter*. The Inverse Fourier Transform of the estimate found in the above equation provides the image in the spatial domain. The power spectrum of the original image is not known in our case and hence, we would need to find an estimate for the same.

Figure 3.2(a) is the original training set image, which is degraded with a Gaussian blur of standard deviation = 3.0 to form the image shown in Figure 3.2(b). Using our estimate of the blur and noise in the degraded image and employing the Trivial Wiener filtering scheme in (3.12) provides the estimate shown in Figure 3.2(c).

3.5.3 Regularized Fourier Domain Wiener Filtering

(3.12) symbolizes the crux of Wiener filtering. But if the PSD of the image Pxx goes to zero, the noise components get amplified and we get very small values of signal components as the result of (3.12). In this thesis, we perform Fourier domain shrinkage (Regularization) in order to attenuate noise components and keep the results within boundary limits.

(a) Original image

(b) Degraded with Gaussian blur of SD = 3.0

(c) Restored image using Trivial Wiener filtering.

Figure 3.2: Trivial Wiener Filtering results.

Regularized Fourier domain Wiener filtering can be expressed thus

$$\tilde{X}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \alpha \frac{P_{\eta\eta}(u,v)}{P_{xx}(u,v)}} \right] Y(u,v) \tag{3.13}$$

As we do not possess knowledge of the ideal image in blind deconvolution, we would need to compute an estimate of the PSD of the ideal image Pxx to be fit into (3.13). We use the PSD of the degraded input itself as the Pxx value in this case. We compute the Wiener estimate for all values of the shrinkage term α.

Once we have a Wiener estimate for every value of α, we can compare and compute the best estimate of the lot. For this, the Wiener estimates for each value of α are convolved with the blur function. The Mean Squared Error between this result and the degraded input image y is then computed as a metric for comparison between different cases.

$$MSE = MSE(\tilde{x} \otimes h, y)$$

That value of α that corresponds to the smallest MSE, say α_1, is then chosen for the final filtering process and the final restored estimate is computed thus.

$$\tilde{X}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + \alpha_1 \frac{P_{\eta\eta}(u,v)}{P_{xx}(u,v)}} \right] Y(u,v)$$

3.5.4 Iterative Wiener Filtering

As can be judged from the requirements for Wiener filtering, an excellent knowledge of the blurring parameters is required. If the estimates of the degradation function and noise

statistics that are used for Wiener filtering are not precise, the restoration function fails to provide satisfactory results.

Estimation of the Power Spectral Density of the original image is needed for Wiener filtering, which in our case is unknown. Hillery and Chin [9] theorized that by using the degraded image as a prototype, an initial estimate of the PSD can be obtained, and subsequently a restored image is obtained, which would be closer to the original image than the degraded image. The PSD of this resulting image is used again for the restoration process. Therefore, this sets in effect, an iterative process where successively restored images are used to obtain better estimates of the PSD, which is subsequently used in the restoration process. But first, we would need an initial estimate obtained using Fourier domain shrinkage and compute the PSD of this estimate.

$$\begin{aligned} \widetilde{X}_{\lambda^f}(f_k) &= Y(f_k)\lambda_k^f \\ &= Y(f_k)\left(\frac{H^*(f_k)}{|H(f_k)|^2 + \Lambda(f_k)}\right) \end{aligned} \tag{3.14}$$

The initial Wiener estimate upon Fourier domain shrinkage is given in (3.14). This step is to be performed for every value of α. As in Sec 3.4.3, the MSE between the convolution of the estimate with the blur function and the degraded input is computed and compared. Using the value of α that corresponds to the smallest MSE, an initial Wiener estimate is computed as in (3.14) and a corresponding PSD estimate is computed.

Because an iterative process is performed so as to converge ultimately to the actual PSD of the image to be restored, this initial estimate for the PSD would be a good beginning.

$$P_{xx} = \left| Y.\frac{H^*(f_k)}{|H(f_k)|^2 + \Lambda(f_k)} \right|^2 \tag{3.15}$$

(3.15) gives the initial estimate of the PSD of the image that would be used to start the iterative process, with that value of α chosen that corresponds to the smallest MSE value.

In summary, the algorithm follows the sequence of steps detailed below.

1. Compute the transfer function of the restoration filter for the given iteration using the Power spectral density computed in the previous iteration.

$$RF = \left[\frac{H^* P_{xx}}{H P_{xx} H^* + P_{\eta\eta}} \right] \tag{3.16}$$

2. Compute an improved estimate of the Power Spectral Density

$$M = Y \,.\, RF \tag{3.17}$$

3. New value of Power Spectral Density to be used in the next iteration

$$Pxx = |M|^2 \tag{3.18}$$

4. Repeat steps 1 to 3 for the set number of iterations. In practice, 10 iterations should provide satisfactory results.

Using the improved PSD estimate P_{xx} of the image to be restored, we now apply Wiener filtering on the degraded image thus.

$$\tilde{X}(u,v) = \left[\frac{H^*(u,v) P_{xx}(u,v)}{P_{xx}(u,v)|H(u,v)|^2 + P_{\eta\eta}(u,v)} \right] Y(u,v) \tag{3.19}$$

The estimate in (3.19) is the result of Fourier domain shrinkage on the degraded image for a given value of the shrinkage parameter α.

As a matter of theoretical interest, all of the processes from (3.14) to (3.19) were repeated for different value of the shrinkage term α. The MSE between the convolution of the estimate in (3.19) with the blur function and the degraded input y is computed for every case of α. The α value that corresponds to the smallest MSE using this iterative process is, in some cases, different from the α value that was obtained if we use a Single-step Regularized Wiener filtering process as in Sec 3.4.3. Hence, it would be best to use this iterative process to compute the shrinkage term α that corresponds to the smallest MSE.

3.5.5 Iterative Wiener Filtering with Additive Correction

The main problem involved with such Iterative filtering is the convergence of the procedure. It is a given that improper estimation of PSD results in undesirable restoration. The relationship between a PSD estimate and the next improved PSD can be summarized in the following equations

$$
\begin{aligned}
P_{xx}(i+1) &= RF(i+1)\, P_{xx}(0)\, RF^{*}(i+1) \\
&= \frac{P_{xx}(i)\, H^{*}}{HP_{xx}(i)H^{*} + P_{\eta\eta}}\, P_{xx}(0)\, \frac{H\, P_{xx}{}^{*}(i)}{HP_{xx}(i)H^{*} + P_{\eta\eta}} \\
&= \frac{P_{xx}(0)\, P_{xx}^{2}(i)\, |H|^{2}}{\left[P_{xx}(i)\, |H|^{2} + P_{\eta\eta} \right]^{2}}
\end{aligned}
\tag{3.20}
$$

The objective of such an iterative Wiener process of updating the Power Spectral density with each epoch is to ensure that the PSD converges to the true value of PSD of

the image. But the estimation of PSD, which is mathematically a covariance estimate, is never precise. An additive correction term can be introduced to get around this problem and achieving convergence.

$$P_{xx}^{+}(i+1) = P_{xx}(i+1) + S(i+1) \tag{3.21}$$

S(i+1) is the correction term. After computing the PSD in the iterative process in (3.17), which is to be used in the next iteration, we add a correction term to it *S*, and use this corrected value of PSD for constructing the restoration filter RF in the next iteration. Previously, the PSD computed used at each step was based on the restored image rather than the ideal image. This correction helps offset this error to a good extent and helps our goal of convergence to the ideal estimate.

In Matrix notations, S – the correction term can be evaluated thus.

$$S(i+1) \quad = \quad P_{xx}^{+}(i)\left[I - \left[RF(i+1)\,H\right]^{t}\right] \tag{3.22}$$

The effectiveness of the correction and a proof of convergence have been presented in [9].

It has been verified in [9] that this correction factor $S(i+1)$ helps lead to the ideal value of the PSD of the image to be restored P_{xx}. Substituting the correction term yields a new update equation for the PSD. The fixed point of the update process can be computed by solving the roots of the equation in (3.23)

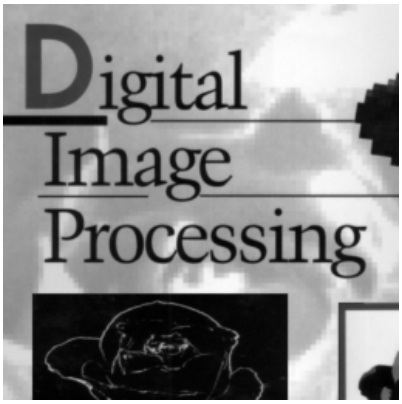
$$P_{xx}^{+}(i+1) - P_{xx}^{+}(i) \quad = \quad \frac{\left|H\right|^{2} P_{xx}^{+\,2}(i)\left[Pxx(0) - \left|H\right|^{2} P_{xx}^{+}(i) - P_{\eta\eta}\right]}{\left[\left|H\right|^{2} P_{xx}^{+}(i) + P_{\eta\eta}\right]^{2}} \tag{3.23}$$

The chosen value of S for each iteration can be expressed thus.

$$S(i+1) = \frac{P_{xx}^{+}(i) P_{\eta\eta}}{\left[|H|^2 P_{xx}^{+}(i) + P_{\eta\eta} \right]} \tag{3.24}$$

In summary, the iterative process with additive correction follows the same process as the one detailed earlier from (3.14) to (3.18). The additive correction term in (3.24) is added to the PSD value computed in (3.18). This result is to be used as the PSD value P_{xx} for the next iteration. Once the final PSD value is computed after the set number of iterations, the Wiener estimate is computed by applying this final PSD estimate in (3.19).

Figure 3.3(a) is an original image that is part of the training set. It is degraded with Gaussian blur of standard deviation = 4.0 to form the degraded version shown in Figure 3.3(b). Upon applying an Iterative Wiener filtering algorithm on this degraded image with our estimate of blur and noise, we get a restored estimate shown in Figure 3.3(c). Figure 3.3(d) shows the result of introducing an Additive correction factor to the Iterative Wiener filtering algorithm, in order to help achieve convergence to the ideal estimate. There does not seem to be a difference between the 2 estimates, in terms of visual quality, but need not be true for other images. In Chapter 5, the results indicate a qualitative difference between estimates obtained using simple Iterative Wiener filtering and Iterative Wiener filtering with additive correction.



(a) Original image

(b) Degraded with Gaussian blur of SD = 4.0

(c) After Iterative Wiener filtering

(d) With additive correction.

Figure 3.3: Iterative Wiener filtering estimates

Chapter 4

Fourier Wavelet Cascaded Deconvolution

4. 1 Deconvolution – A Cascaded Approach

Neelamani et al [11] proposed a hybrid algorithm ForWaRD, the essence of which was to perform scalar shrinkage in the Fourier domain followed by a similar operation in the wavelet domain, unlike the strictly Fourier domain shrinkage discussed in Chapter 3. The advantage of such an approach is manifold. While the shrinkage in the frequency domain would help improve the efficiency of noise representation, the wavelet shrinkage helps make good use of the domain's economical representation of signals - in our case, images [12]. The Mean Square Error is used as an optimization criterion to derive the best trade-off between the shrinkage in the Fourier and Wavelet domains. It has been established that a small amount of Fourier shrinkage would suffice for signals with economical wavelet representations.

A block diagram illustrating the working of such a method follows in Figure 4.1.

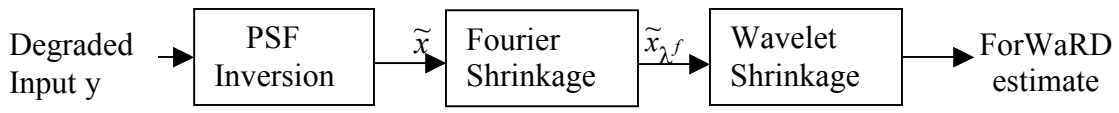


Figure 4.1 Fourier-Wavelet Cascaded Deconvolution model

The need for a twin approach – involving both Fourier and Wavelet domains is now obvious. The wavelet domain cannot economically represent noise and the Fourier domain cannot economically represent signals, as much as it can represent noise. As can be seen from Figure 4.1, after operator inversion on the degraded image, a Fourier domain based regularized deconvolution is performed, much like the Wiener regularized approach discussed in Chapter 3. Once an estimate is obtained from this approach, a wavelet domain based regularized deconvolution is performed, not unlike the Wiener based approach for the Fourier domain. Only this time, the deconvolution operation itself is performed in the Wavelet domain. In order to obtain the shrinkage value that would yield best results, the MSE estimate between the original image and the result after the Wavelet domain filtering is computed to be used as an optimization parameter in order to obtain a good fit for the shrinkage term.

4.2 Experiment Outline

Neelamani et al had knowledge of the original image (the image to be restored) and the blurring function that resulted in the degraded image. In this thesis, we consider a blind deconvolution problem initially. After computing the PSF and an estimate of the noise variance in the original image, we employ these parameters in a deconvolution set-up. In this thesis, we seek to employ robust techniques to estimate the Fourier domain estimate before passing on to the wavelet domain stage. We also explore the effectiveness of soft-thresholding among other de-noising approaches in the wavelet domain.

4.3 The Experiment

Given that we have a degraded image y that needs to be restored to x, their relationship can be expressed thus

$$Y(f_k) = H(f_k)X(f_k) + \Gamma(f_k) \tag{4.1}$$

Y and X are the respective Discrete Fourier Transforms of *y* and *x*;
H and Γ are the respective Discrete Fourier Transforms of the impulse response *h* of the degradation system and the noise variance η.

A trivial estimate of the image to be restored can be computed simply by operator inversion.

$$\tilde{x}(n) \cong H^{-1} y(n) = x(n) + H^{-1}\gamma(n) \tag{4.2}$$

The problem of *H* being ill-conditioned makes us seek better solutions to estimate the image to be restored.

4.3.1 Noise Computation

The noise variance, as was discussed in Chapter 3, can be reliably estimated by computing the median absolute value of the highest resolution wavelet coefficients and dividing it by a constant [3].

$$\sigma = \frac{median\left(\left| W(1,x) \right|\right)}{0.6745} \tag{4.3}$$

W (1,x) indicates the wavelet coefficients in the highest resolution band or the first scale.

We order the finest scale wavelet coefficients and compute the median of this array and divide it by a constant as shown in (4.3) to obtain the noise variance value for the image under study.

4.3.2 Initial Experimental Set-up

The problem set-up in the Fourier domain is expressed in (4.1). A simple inversion operation in the Fourier domain, which leads to a trivial estimate, can be expressed thus

$$\widetilde{X}(f_k) = X(f_k) + \frac{\Gamma(f_k)}{H(f_k)} if \left|H(f_k)\right| > 0,$$
$$\widetilde{X}(f_k) = 0, otherwise \tag{4.4}$$

$\tilde{x}$ is the DFT of $\tilde{x}$, which is the estimate of the image to be restored.

It can be seen from (4.4) that given values of f_k at which the blur function H is very small i.e. as it approaches zero, the noise components denoted by Γ are amplified significantly. As can be reasoned, this would result in poor Signal to Noise Ratio values and would result in a noisy estimate.

This creates the need for shrinkage. Regularized deconvolution [9] has been explored before and has proven effective in improving SNR values of estimates. In this thesis, we apply a regularization term denoted by $\Lambda(f_k)$ in the shrinkage parameter λ_k^f

$$\lambda_k^f = \frac{\left|H(f_k)\right|^2}{\left|H(f_k)\right|^2 + \Lambda(f_k)} \tag{4.5}$$

Note that the regularization term is always greater than or equal to zero. As in Tikhonov regularized deconvolution [10], we set it to be a constant that can be expressed thus

$$\Lambda(f_k) = \alpha.\frac{N.\sigma^2}{\left\|\hat{x} - \bar{\hat{x}}\right\|_2^2} \tag{4.6}$$

N is the size of the signal, in our case, the total number of pixels in the image; σ is the noise variance; $\hat{x}$ is the estimate obtained on the Fourier domain operator inversion of the degraded image y by h and $\bar{\hat{x}}$ is its mean value. In practice, we use an array of values for α, which are logarithmically spaced.

4.3.3 Fourier Domain Shrinkage

Applying this shrinkage in the Fourier domain to the estimate $\widetilde{X}(f_k)$ in Wiener form,

$$\widetilde{X}_{\lambda^f}(f_k) = \widetilde{X}(f_k)\lambda_k^f$$

$$= \widetilde{X}(f_k)\left(\frac{H^*(f_k)}{|H(f_k)|^2 + \Lambda(f_k)}\right) \tag{4.7}$$

This provides us with an initial estimate of Wiener estimate in the Fourier domain. We compute this initial Fourier domain Wiener estimate in (4.7) for each value of the shrinkage term α and apply Wavelet domain shrinkage on each of these estimates.

We explore, from the point of experimental interest, to obtain the desired shrinkage term α after Iterative Filtering in the Fourier domain. Specifically, this entails performing Iterative Wiener Filtering in the Fourier domain in order to get a Fourier domain estimate for each value of the shrinkage term. We then apply Wavelet domain shrinkage on each of these estimates to obtain a final estimate. The shrinkage term corresponding to the smallest MSE value is then chosen to be used in the final Fourier-Wavelet Cascaded deconvolution setup.

4.3.4 Need for Wavelet Domain Shrinkage

The Fourier domain is advantageous in the sense that it provides the most economical representation of colored noise and the maximum colored noise energy is captured using a set number of coefficients. The total MSE estimate in the Fourier set-up could be possibly lower bounded by noise variance. However, as much as it can represent noise economically, the Fourier domain does not provide economical representations for images with edge-type singularities as the energy of such features spreads over many Fourier coefficients. As a result, the MSE is actually unsatisfactory even with the best

possible regularization. Wavelets, on the other hand, can represent signals with singularities economically. The objective of Wavelet domain regularized deconvolution would be to leverage the wavelet domain's economical representation of edge components to a noisy input in which the noise is already represented very effectively.

4.3.5 Wavelet Domain Set-up

Initially, we need to select the wavelet experimental set-up. It should be noted that if we use a Discrete Wavelet Transform, the resulting wavelet coefficients do not possess shift-invariance capabilities. Hence, we would need to use a shift-invariant transform such as Steerable Pyramids [13] or employ a Redundant Discrete Wavelet Transform. In this thesis, we utilize the redundant, shift-invariant DWT in order to obtain shift-invariant estimates by averaging over all possible shifts. A Wavelet domain based Wiener filter is implemented for performing Regularized Deconvolution. Before the actual Wiener filtering, de-noising methods such as Hard and Soft Thresholding are also implemented on the Fourier domain Wiener estimate to be used in the wavelet shrinkage process.

4.3.6 Wavelet Domain Deconvolution

In practice, a 6-coefficient Dauebchies filter based mother wavelet is used for computing the Redundant Discrete Wavelet Transform (RDWT) of the Fourier domain estimate $\tilde{x}_{\lambda^f}$.

$$\tilde{w}_{j,l;\lambda^f} = RDWT(\tilde{x}_{\lambda^f}) \tag{4.8}$$

We seek to shrink the wavelet transform of the Fourier estimate, $\tilde{w}_{j,l;\lambda^f}$, using a regularization term $\lambda^w_{j,l}$. The RDWT of the spatial domain form of the Regularization Inversion filter *r* is computed for estimating noise variance at different wavelet scales using *de-noising* techniques, the most efficient of which are detailed below.

4.3.6.1 Hard Thresholding

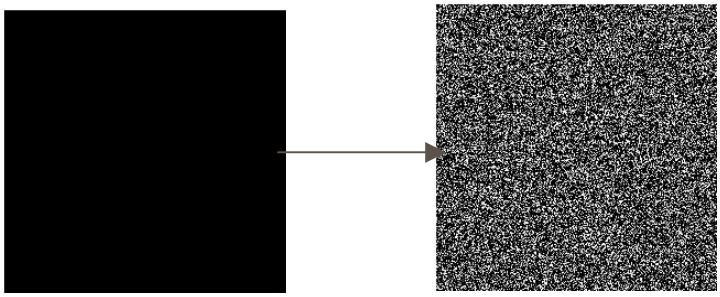
The essence of Hard Thresholding [14] consists of first stipulating a threshold level for the given wavelet scale T_j and then setting to zero all wavelet coefficients which have an absolute value that is lower than this threshold.

$$T_j = K \sigma_j \tag{4.9}$$

j indicates the wavelet scale and σ_j is the standard deviation of the noise at scale j. K is a constant referred to as Threshold Factor and is generally chosen to be equal to 3.0. In the case that that the wavelet transform algorithm is an energy-normalized one, then the standard deviation of noise at all scales is equal to the standard deviation of noise in the image under consideration.

In order to estimate the standard deviation of noise at various wavelet scales, it is best to simulate a zero value signal or a black image and corrupt it with Gaussian noise with a standard deviation of unity (Refer Figure 4.2).

On decomposing this zero intensity signal or image into various resolution levels using a wavelet transform, the standard deviation of the noisy signal at each scale σ_j^e is computed.



(a) Black image (b) After corrupting with noise

Figure 4.2: Corrupting a zero intensity image with Gaussian noise

By this, we obtain an idea of the variation in noise level across scales. Using the properties of the wavelet transform then helps us relate this value to the actual standard deviation of noise in the case of our signal or image [15].

$$\sigma_j = \sigma \sigma_j^e \tag{4.10}$$

σ is the standard deviation of noise in the image under consideration. This, as has been discussed in Sec 4.3.1, can be reliably estimated by computing the median absolute value of the wavelet coefficients in the finest scale and dividing it by 0.6745.

Mathematically, Hard Thresholding can be summarized thus,

$$\widetilde{w}_{j,k} = \begin{cases} w_{j,k} & if \ |w_{j,k}| \geq T_j, \\ 0 & if \ |w_{j,k}| < T_j. \end{cases} \tag{4.11}$$

4.3.6.2 Soft Thresholding

In Soft Thresholding [7], a threshold for each wavelet scale is stipulated as in Hard Thresholding. But for wavelet coefficients at a given scale whose absolute value is greater than the threshold, the difference between the coefficient and the threshold in computed, unlike Hard Thresholding, where the value of the wavelet coefficient is fully retained if greater than the threshold.

$$\widetilde{w}_{j,k} = \begin{cases} \left(|w_{j,k}| - T_j\right) & if \ |w_{j,k}| \geq T_j, \\ 0 & if \ |w_{j,k}| < T_j. \end{cases} \tag{4.12}$$

4.3.6.3 Donoho Approach

The main difference between the Donoho approach [7] and the previous thresholding approaches is the selection of threshold levels for each wavelet scale. Unlike previous

approaches where the threshold is selected based on a constant Threshold Factor of 3, the threshold in this case is selected based on the number of pixels in the image.

$$T_j = \sqrt{2\,log(n)}\,\sigma_j \tag{4.13}$$

One disadvantage of this approach is that it could overly smooth the data. The turn-around for this problem is to apply the *minimax* criterion discussed in [16].

Once this threshold is computed for each wavelet scale, Soft Thresholding is chosen to shrink the wavelet coefficients in the same way as before.

$$\widetilde{w}_{j,k} = \begin{cases} \left(|w_{j,k}| - T_j\right) & if\ |w_{j,k}| \geq T_j, \\ 0 & if\ |w_{j,k}| < T_j. \end{cases} \tag{4.14}$$

4.3.6.4 Multi-resolution Wiener Filtering

The crux of this method consists of computing the product between the wavelet coefficients for a given scale and a ratio of the signal variance at that scale to the sum of the signal and noise variance at the same wavelet scale [17].

$$\begin{aligned} \alpha_j &= \frac{S_j}{S_j + N_j} \\ \widetilde{w}_{j,k} &= w_{j,k}\,\alpha_j \end{aligned} \tag{4.15}$$

S_j is the variance of the signal and N_j is the noise variance for wavelet scale j.

In the case that signal variance is unknown, as in our case where the signal under study contains noise, the difference between the variance of the data and the variance of the noise is used.

4.3.7 Wiener Wavelet Filtering

Once the de-noised estimate is obtained, the wavelet shrinkage is implemented using a Wiener Filtering algorithm in the wavelet domain [18]. But first, the RDWT of the denoised estimate and of the Fourier domain estimate is computed with a different number of vanishing moments [18]. The wavelet transform of the regularization inversion filter *r* is computed and the noise variance at various wavelet scales is obtained using the same principle as detailed in the section in Hard Thresholding (Sec 4.3.6.1). (i.e.) $\sigma_j = \sigma\,\sigma_j^e$

σ is the standard deviation of noise in the image under consideration; σ_j^e is the standard deviation of noise at scale *j* for the simulated signal (zero signal – additive noise with standard deviation of 1); σ_j is the standard deviation of the noise at scale *j* for the image under consideration.

Wavelet domain Wiener filtering can be mathematically expressed thus

$$\lambda_{j,l}^{w} = \frac{\left|w_{j,l}\right|^2}{\left|w_{j,l}\right|^2 + \sigma_j^2} \tag{4.16}$$

The Wavelet domain Wiener filtering improves on the MSE performance of Thresholding approaches discussed earlier by using Wiener estimation on each wavelet coefficient. The actual wavelet coefficients needed to this estimation, are as such unknown, in our case. Hence, we use the wavelet coefficients obtained after de-noising using Hard, Soft or Donoho Thresholding to compute the shrinkage term $\lambda_{j,l}^{w}$ in (4.16).

In our case, (4.16) can be written as

$$\lambda^{w}_{j,l} = \frac{\left|\tilde{w}^{1}_{j,l}\right|^{2}}{\left|\tilde{w}^{1}_{j,l}\right|^{2} + \sigma^{2}_{j}} \tag{4.17}$$

$\tilde{w}^{1}_{j,l}$ is the RDWT of the spatial domain form of the de-noised estimate. An RDWT involving a 2 coefficient Daubechies filter is used to obtain $\tilde{w}^{1}_{j,l}$.

Once the shrinkage is estimated, the final estimate in the wavelet domain can be expressed thus.

$$\tilde{x}_{\lambda^{w}_{j,l}} = IRDWT\left(w^{1}_{j,l}\ \lambda^{w}_{j,l}\right) \tag{4.18}$$

$w^{1}_{j,l}$ is the RDWT of the spatial domain form of the Fourier domain final estimate and is obtained using a 2 coefficient Daubechies filter. It is to be noted that a 6-coefficient filter was used for computing the wavelet transform of the Fourier domain estimate during denoising and it is necessary that filters with different coefficients be used during the two processes [18].

The final estimate of this Fourier-Wavelet cascaded approach in given in (4.18) for one case of the shrinkage term α. This entire Fourier-Wavelet shrinkage approach is to be repeated for every value of α. As in Chapter 3, we use the Mean Squared Error between the convolution of the estimate in (4.18) with the blur function and the degraded input y as a performance metric to compare results for different cases of α. The value of shrinkage term α that corresponds to the smallest MSE value is then used in the final filtering process.

4.3.8 Final Filtering Mechanism

The value of α that corresponds to the smallest MSE is used to compute an initial Fourier domain estimate from which a corresponding PSD value is computed

$$P_{xx} = \left| Y . \frac{H^*(f_k)}{|H(f_k)|^2 + \Lambda(f_k)} \right|^2 \quad (4.19)$$

In this thesis, the *Iterative Wiener filtering with Additive correction* algorithm is employed to compute an improved PSD estimate. It can be summarized thus.

Step 1: Compute the transfer function of the restoration filter for the given iteration using the Power spectral density computed in the previous iteration.

$$RF = \left[\frac{H^* P_{xx}}{H P_{xx} H^* + P_{\eta\eta}} \right] \quad (4.20)$$

Step 2: Compute an improved estimate of the Power Spectral Density

$$M = Y . RF \quad (4.21)$$

Step 3: Compute the additive term that provides offset correction

$$P_{xxY} = \frac{P_{xx} . P_{\eta\eta}}{P_{hh} P_{xx} + P_{\eta\eta}} \quad (4.22)$$

Step 4: New value of Power Spectral Density to be used in the next iteration

$$Pxx = P_{xxY} + |M|^2 \quad (4.23)$$

Repeat steps 1 through 4 for the desired number of iterations.

Using the improved PSD estimate P_{xx} of the image to be restored, we now apply Wiener filtering on the degraded image thus.

$$\tilde{X}(u,v) = \left[\frac{H^*(u,v)P_{xx}(u,v)}{P_{xx}(u,v)|H(u,v)|^2 + P_{\eta\eta}(u,v)} \right] Y(u,v) \quad (4.24)$$

The estimate in (4.24) is the result of Fourier domain shrinkage on the degraded image for the chosen value of the shrinkage parameter α.

For comparison purposes, Iterative Wiener filtering aside, a single step LTI Wiener deconvolution process is also implemented. In this case, the transfer function of the Regularization based Inversion filter *R* is first computed.

$$R = \left(\frac{H^*(f_k)}{|H(f_k)|^2 + \Lambda(f_k)} \right) \quad (4.25)$$

Once the inversion filter is computed, the Fourier domain Wiener estimate can be arrived at.

$$X_f = R \,.\, Y \quad (4.26)$$

The Inverse Fourier Transform of the Fourier estimate from (4.26) provides the Wiener estimate in the spatial domain.

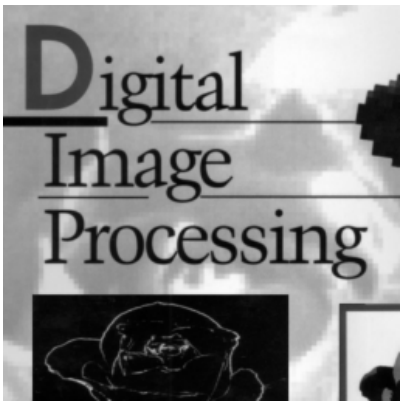
On this Wiener estimate obtained in (4.24) – Iterative Wiener Filtering result and (4.26) – Single step Wiener Filtering, we apply wavelet domain shrinkage as detailed in Sec 4.3.6 and Sec 4.3.7 i.e. the de-noised estimate of this Fourier domain estimate is first computed using Hard-Thresholding or Soft-Thresholding. This de-noised estimate is then used to construct the Wavelet domain Wiener filter for which the noise variances are computed using the Regularization filter R. Applying Wavelet domain Wiener filtering on each wavelet coefficient of the RDWT of the Fourier domain Wiener estimate completes the process of this cascaded deconvolution approach.

Figure 4.3(a) is an original image that is part of the training set. It is degraded with Gaussian blur of standard deviation = 4.0 to form the degraded version shown in Figure 4.3(b). Upon applying an Iterative Wiener filtering algorithm on this degraded image in the Fourier domain with our estimate of blur and noise, we get a restored estimate shown in Figure 4.3(c). Figure 4.3(d) shows the result of applying a Fourier-Wavelet Cascaded deconvolution to restore the image. The first part of this restoration process consists of employing the Iterative Wiener filtering with additive correction algorithm, in order to help achieve convergence to the ideal estimate. The next step in the restoration process consists of applying Wiener filtering on this Fourier domain estimate in the Wavelet domain. There is a qualitative difference between the estimates shown in Figures 4.3(c) 4.3(d). The Fourier-Wavelet cascaded deconvolution estimate appears visually robust compared to the Fourier domain estimate. In Chapter 5, we discuss the results of applying these deconvolution algorithms on other test images and observe the superior performance of the Fourier-Wavelet cascaded deconvolution over the Iterative Wiener filtering algorithm in the Fourier domain.

4.4 Technical Summary

The first step in this Fourier-Wavelet cascaded approach is to obtain an initial estimate of image to be restored X by operator inversion i.e. Perform operator inversion on the degraded image Y by inverting the PSF function H. (Note: All parameters in Fourier domain). During this inversion process, at small values of H, the noise components in the Fourier domain are considerably amplified.

In order to attenuate these noisy components, shrinkage is employed in the Fourier domain, by choosing a small value for the regularization parameter that determines the shrinkage term. The parameter α that determines the value of the regularization parameter is varied in a logarithmic scale, so that the value of α for which minimum MSE results can be noted and used in the final algorithm implementation.



(a) Original image

(b) Degraded version

(c) Fourier domain filtering

(d) Fourier-Wavelet Cascaded filtering

Figure 4.3: Fourier and Fourier-Wavelet Cascaded filtering estimates

The crux of this shrinkage process is to attenuate the amplified noise components while retaining the signal components to the best extent possible. By choosing different values of α and observing the MSE results, the best value of α that makes up the regularization parameter and in effect, the shrinkage term is noted and employed ultimately in the shrinkage algorithm.

Once the Fourier shrinkage is completed, the Redundant Discrete Wavelet Transform of the Fourier domain estimate is computed. Because of the properties of the Wavelet transform, the estimate is represented economically in the wavelet domain. In order to apply Wiener filtering in the wavelet domain, the shrinkage term is to be first computed. Ideally, the shrinkage term must consist of the wavelet coefficients of the ideal image, which is unknown in our case. This case is analogous to the Wiener filtering in the Fourier domain, in which we would ideally require the Power Spectral density of the actual ideal image, which is unknown. In the wavelet domain, a variety of Thresholding approaches such as Hard, Soft and Donoho thresholding exist for denoising the estimate to get an improved estimate. In practice, we apply these denoising methods to get an improved estimate of wavelet coefficients, using a threshold factor of 3. The shrinkage term for a given scale and location is a ratio of the square of the denoised wavelet coefficients for the corresponding scale and location to the sum of the denoised wavelet coefficients and the noise variance, the latter for the same scale.

Chapter 5

Results and Conclusion

In this chapter, the results of our experiment on PSF estimation and subsequent deconvolution are detailed.

5.1 PSF Estimation

As has been already discussed, the first part of this work deals with the blur estimation. Specifically, we set out to compute the standard deviation of the Point Spread Function, which has been assumed to be a Gaussian, in this case. Using Mallat's theory, a constant absolutely proportional to the Lipschitz exponent is computed for every wavelet trace and the mean value of this exponent is computed for all traces in the image under consideration. It can be summarized that the wavelet evolution across scales thus depends on the following factors [2].

1. The regularity of the original underlying signal.
2. The properties of the wavelet basis functions used.
3. The blur of the signal at the given position.

Hence, we set to compute an estimate of the blur as we know the above two factors. The statistical mean of the exponent for many training images degraded with the same blur value is computed. In effect, the relationship between the mean Lipschitz exponent and the standard deviation of the blur can be expressed thus

$$\sigma = a\ exp(bx) \tag{5.1}$$

x in this case, is the mean value of the Lipschitz exponent.

As was detailed in Chapter 2, we consider a set of training images that are devoid of blur (i.e.) the Gaussian Point Spread Function associated with these images is assumed to have a standard deviation equal to zero. In practice, we apply Gaussian blur with a standard deviation ranging from 1.0 to 6.0 for each of these training images. The average of the Mean Lipschitz exponent for each case of standard deviation is computed using these degraded images. As we have knowledge of the mean Lipschitz exponent for each case of standard deviation, two of the variables in (5.1) - σ and x are known. Using Least Squares Fitting algorithm, the values of a and b in (5.1) are computed.

Now, given the best-fit values obtained for a and b, we compute the standard deviation σ for each of the training set images for each value of their Mean Lipschitz exponent. The Mean Lipschitz exponent for every image for each value of applied blur is computed using the PSF Estimation algorithm detailed in Chapter 2.

Figure 5.1 shows the blur estimate that was obtained using the exponential fitting in (5.1).

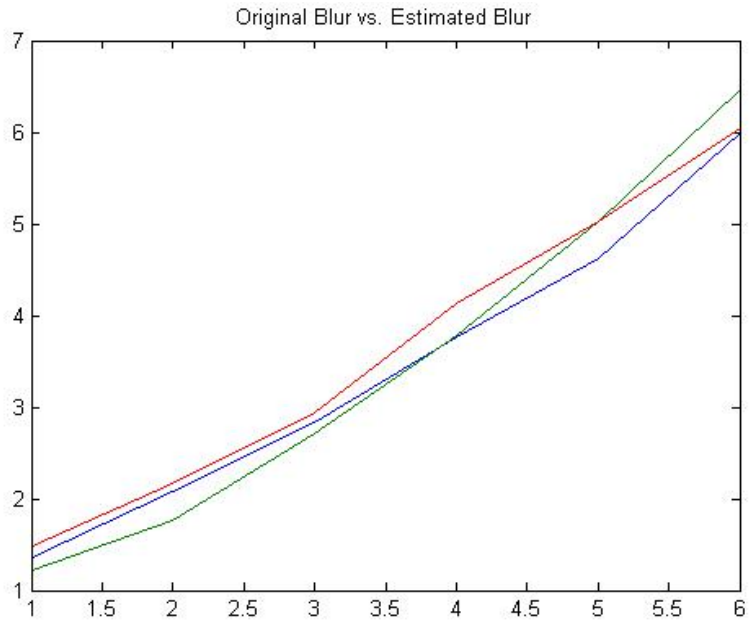


Figure 5.1: Applied blur vs. estimated blur for training images using exp1 fitting

As can be seen from Figure 5.1, the estimated values of standard deviation fall within the targeted range but are not satisfactory in that they do not fall within permissible limits of the original value. Ideally, the results of this Image Restoration exercise would be best served only if the accuracy of estimation is satisfactory. We try to improve on the obtained results by modifying (5.1) thus

$$\sigma = a\ exp(bx) + c\ exp(dx) \tag{5.2}$$

Given the average center of gravity of the Lipschitz exponent for each value of standard deviation, we employ Least Squares Fitting algorithm with this particular exponential case to obtain best-fit values for a and b. Using these values of a and b and applying it in (5.2) for every CG value of the Lipschitz exponent for each of the 3 training images, we obtain estimates of the standard deviation for each value of applied blur for the 3 images.

As can be seen from Figure 5.2, the accuracy of the estimated values is improved with the introduction of this additional exponential term for obtaining best-fit values.

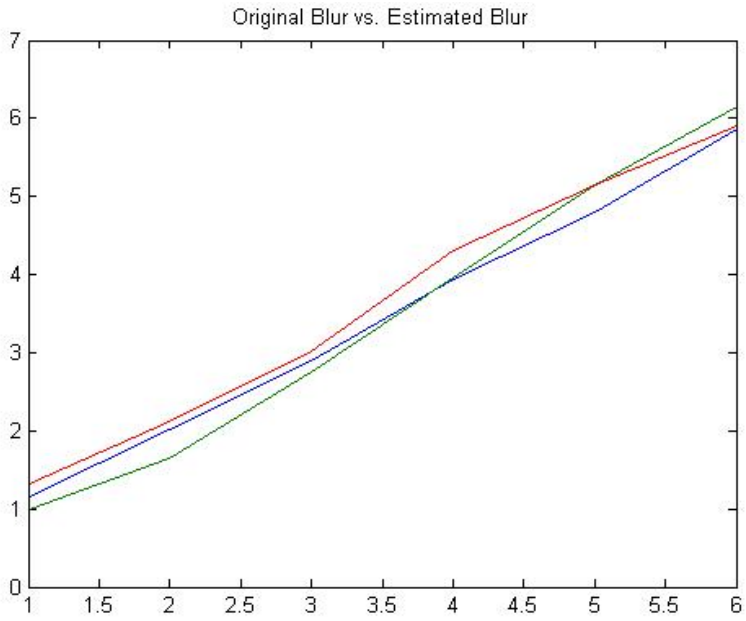


Figure 5.2: Applied blur vs. estimated blur for training images using exp2 fitting

Table 5.1 Estimated values of standard deviation for training images

Training set Image 1		**Training set Image 2**		**Training set Image 3**	
Original SD	***Estimated SD***	***Original SD***	***Estimated SD***	***Original SD***	***Estimated SD***
1.0	1.16	1.0	0.99	1.0	1.30
2.0	2.01	2.0	1.65	2.0	2.12
3.0	2.89	3.0	2.76	3.0	3.02
4.0	3.95	4.0	3.96	4.0	4.31
5.0	4.80	5.0	5.15	5.0	5.14
6.0	5.87	6.0	6.14	6.0	5.91

Original SD – Standard Deviation of blur applied.
Estimated SD – Standard Deviation of blur estimated using our algorithm.

A tabulation of the estimated values of standard deviation for 3 of the training set images can be found in Table 5.1. Once we obtain good fit values for a and b, we seek to use these values to estimate the blur in a test image. But initially, we would need to apply Gaussian blur of standard deviation ranging from 1.0 to 6.0 on the test image and compute the mean Lipschitz exponent for each case of standard deviation.

Applying the best-fit values obtained using these training images in (5.2) for the center of gravity of Lipschitz exponent obtained for the test image, the blur in the test image at each value is computed by fitting in the values of x (the mean Lipschitz exponent for a given value of standard deviation) and constants a and b in (5.2). The plot in Figure 5.3 shows the value of the estimated value of standard deviation for each value of standard deviation of the applied Gaussian blur.

Figure 5.3 shows the blur estimates obtained for the degraded test images when utilizing the exponential relationship in (5.2).

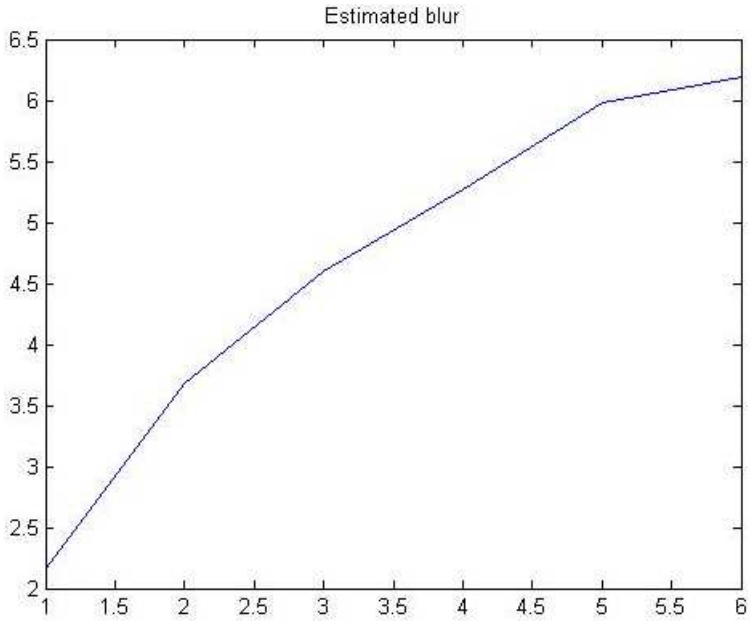


Figure 5.3 Actual Blur vs. Estimated Blur for the test image

As can be seen from Figure 5.3, there is a considerable difference between the standard deviation of the applied value of the blur and the estimated value of the standard deviation computed using our algorithm. This is because the computed value is an estimate of both the applied blur to the testing image as well as the initial blur that already existed in the testing image. We hence try to retain only that portion that corresponds to the already existing blur in the image and remove the applied blur portion.

The actual values of standard deviation of the applied blur and the estimated blur are detailed in Table 5.2. After using (5.2) to obtain an estimate of both the blur applied to the test image as well as the blur that already existed in the test image, we try to compute that value of blur that was already existing in the test image using (5.3).

$$\sigma_{eff} = \sqrt{\sigma^2 - \sigma_{blEst}^2} \tag{5.3}$$

σ_{blEst}^2 is the estimated value of standard deviation σ_{bl} , found when employing the training set images.

Table 5.2 Standard Deviation values of applied blur and estimated blur for test image.

Original SD	***Estimated SD***
1.0	2.15
2.0	3.67
3.0	4.59
4.0	5.27
5.0	5.97
6.0	6.18

It is best to compute the latter for each value of the applied blur and compute the center of gravity for the values. In practice, for the given test image, an effective value of 2.8 is obtained for the standard deviation of its blur.

5.2 Deconvolution Results

The results in this section are analyzed in the following domains and respective set-ups

1. Fourier Domain Wiener Filtering
    - Iterative Wiener Filtering.
    - Iterative Wiener Filtering with Additive Correction factor.
2. Fourier-Wavelet Cascaded Wiener Filtering
    - Iterative Wiener Filtering.
    - Iterative Wiener Filtering with Additive Correction factor.

The results for each of the 4 cases above are discussed together with the performance in each of the cases for different number of iterations for both training and test set images.

5.2.1 Fourier Domain Wiener Filtering

The estimates obtained using Iterative Wiener filtering in the Fourier domain for a training set image are shown in Figure 5.4.

(a) Original Image

(b) Degraded Image

(c) Wiener Filtered, 10 iterations

(d) Wiener Filtered, 100 iterations

(e)Additive Correction, 10 iterations

(f) Additive Correction, 100 iterations

Figure 5.4: Fourier Domain Iterative Wiener Filtering for training image

Figure 5.4(a) shows the original *lena* image that was degraded with a Gaussian blur of standard deviation 3.0 to form Figure 5.4 (b). The PSF Estimation algorithm detailed earlier computed a blur with a standard deviation of 2.89 for this case. After computing the noise in the image using a median estimator on the finest scale coefficients, Fourier domain Wiener Filtering is applied to produce the other results shown in Figure 5.4. Figure 5.4(c) is the result of Iterative Wiener Filtering for 10 iterations and Figure 5.4(d) is the result of the same process for 100 iterations. Adding a correction factor to the Iterative Wiener Filtering process helps the process of convergence towards desired results, at least in a theoretical sense. But as can be seen in figures 5.4(e) and 5.4(f), which show the results of this process for 10 and 100 iterations, the quality of the output becomes worse for higher number of iterations. The Mean Squared Error between the estimate obtained and the original image to be restored is computed. The variation of this MSE with the number of iterations used in the Iterative Wiener Filtering process is illustrated in Figure 5.5.

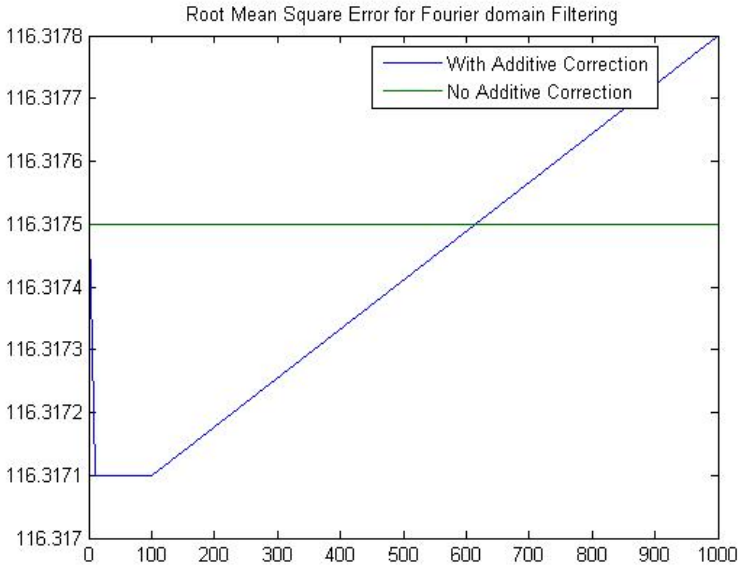


Figure 5.5: RMSE between original image and estimate vs. number of iterations

5.2.2 Fourier Wavelet Cascaded Deconvolution

Figure 5.6 shows the variation in the Root Mean Square Error between the original image and the estimate obtained using Fourier-Wavelet Cascaded Deconvolution with the number of iterations. The estimates obtained using Fourier-Wavelet Cascaded Deconvolution algorithm for a training set image are shown in Figure 5.7.

Figure 5.7(a) shows the original *lena* image that was degraded with a Gaussian blur of standard deviation 3.0 to form Figure 5.7(b). The PSF Estimation algorithm detailed earlier computed a blur with a standard deviation of 2.89 for this case. After computing the noise in the image using a median estimator on the finest scale coefficients, Fourier domain Wiener Filtering followed by Wavelet domain Filtering is applied to produce the other results shown in Figure 5.7. Figure 5.7(c) is the result of Iterative Wiener Filtering for 10 iterations and Figure 5.7(d) is the result of the same process for 100 iterations. Adding a correction factor to the Iterative Wiener Filtering process helps the process of convergence towards desired results, at least in a theoretical sense.

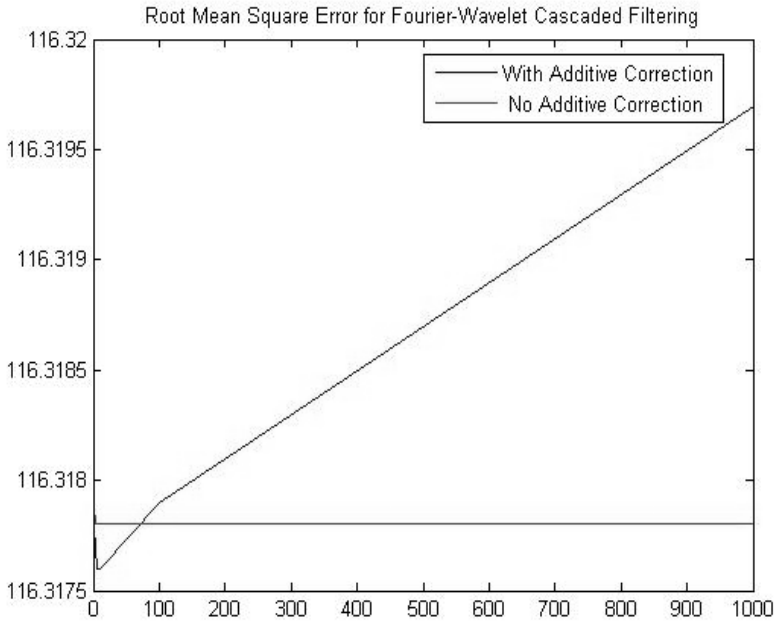


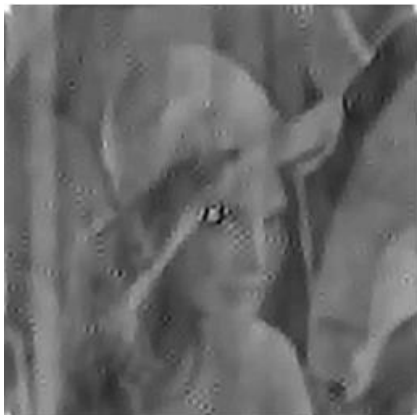
Figure 5.6: RMSE between original image and estimate for Cascased Deconvolution

(a) Original Image

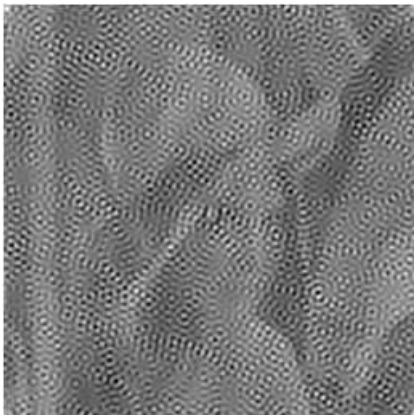
(b) Degraded Image

(c) 10 iterations

(d) 100 iterations



(e) Additive Correction, 10 iterations



(f) Additive Correction, 100 iterations

Figure 5.7: Fourier-Wavelet Cascaded Filtering for the training set image

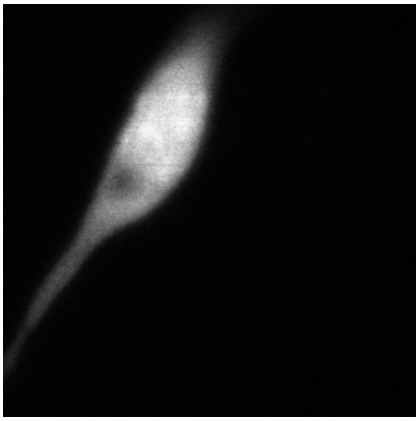
5.3 Deconvolution on the Test Image

The deconvolution estimates obtained for the test set images are shown in this section. The results of Fourier Domain Iterative Wiener Filtering for the test set image are shown in Figure 5.8.

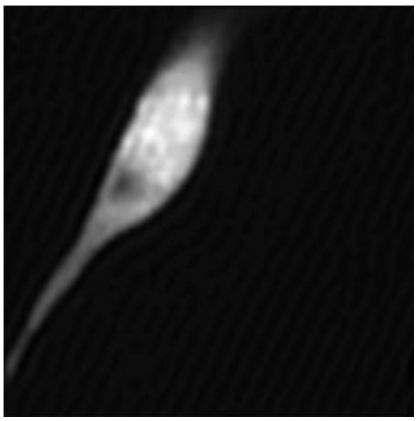
Figures 5.8 (b), (c) and (d) are the results of applying Fourier domain Iterative Wiener Filtering to the original image shown in Figure 5.8(a) for 5, 100 and 1000 iterations respectively. It can be observed that the image obtained for 1000 iterations appears to be visually better than the one obtained after 5 and 100 iterations. The image in Figure 5.8(e) is a close-up version of the original image in Figure 5.8(a) while the image in Figure 5.8(f) is a close-up version of the output of the Fourier domain Filtering process after 100 iterations, shown in Figure 5.8(c). As was detailed earlier, the PSF Estimation algorithm provided an output of 2.8 as the standard deviation of blur in this test image and hence this was the value used for deconvolution.

The images in Figure 5.9 show the results of this deconvolution algorithm for other values of standard deviation. As can be observed from Figure 5.9, the deconvolution estimate when using a blur with a standard deviation of 2.8 produces a crisper image (Figure 5.9(c)) than the ones obtained when using standard deviation values of 1.5 (Figure 5.9(a)) and 4.5 (Figure 5.9(c)).

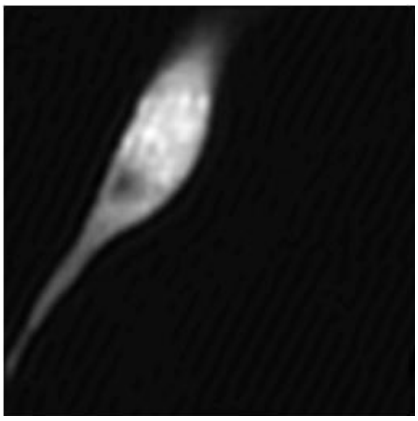
Figure 5.10 shows results obtained using Iterative Wiener Filtering in the Fourier domain. There was a qualitative difference between Iterative Wiener Filtering and the same along with Additive Correction, which was explained using the training set image. In the next figure, we analyze images obtained using Iterative Wiener Filtering with Additive Correction in the Fourier domain.



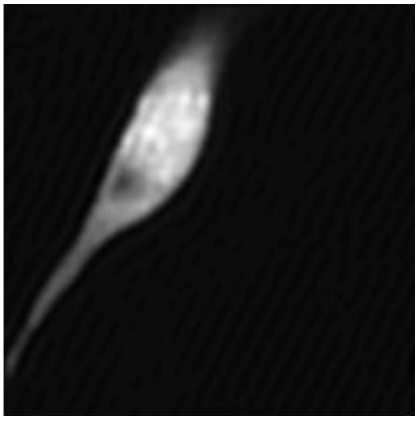
(a) Original Image



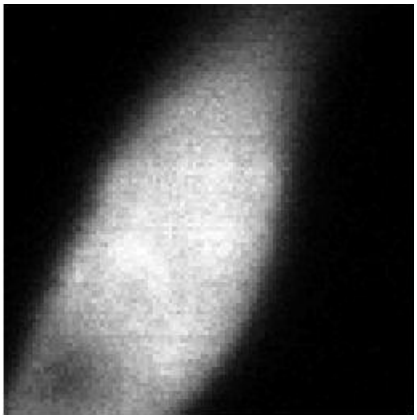
(b) After 5 iterations



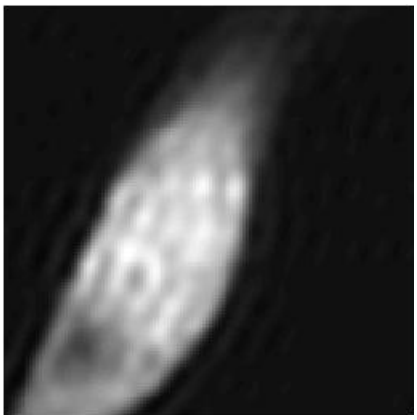
(c) After 100 iterations



(d) After 1000 iterations

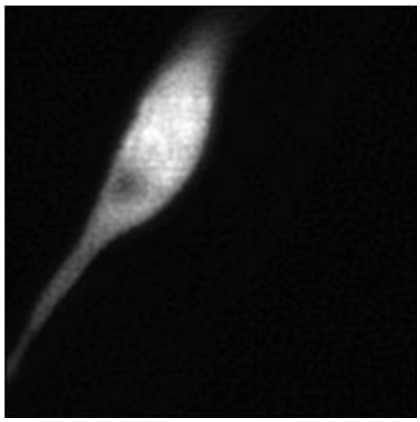


(e) Zoom-up of (a)

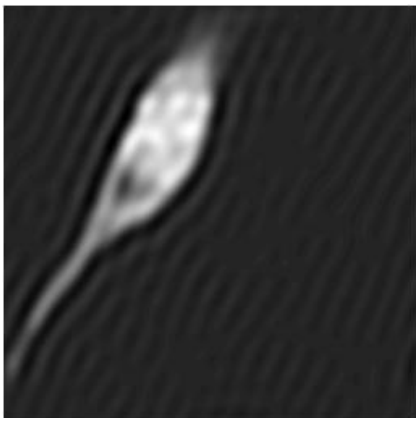


(f) Zoom-up of (c)

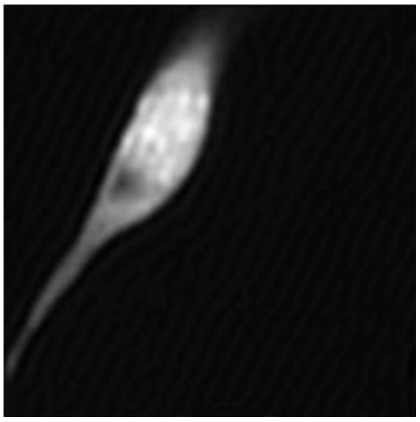
Figure 5.8: Fourier Domain Iterative Wiener Filtering for the test set image



(a) *SD* of Gaussian blur = 1.5

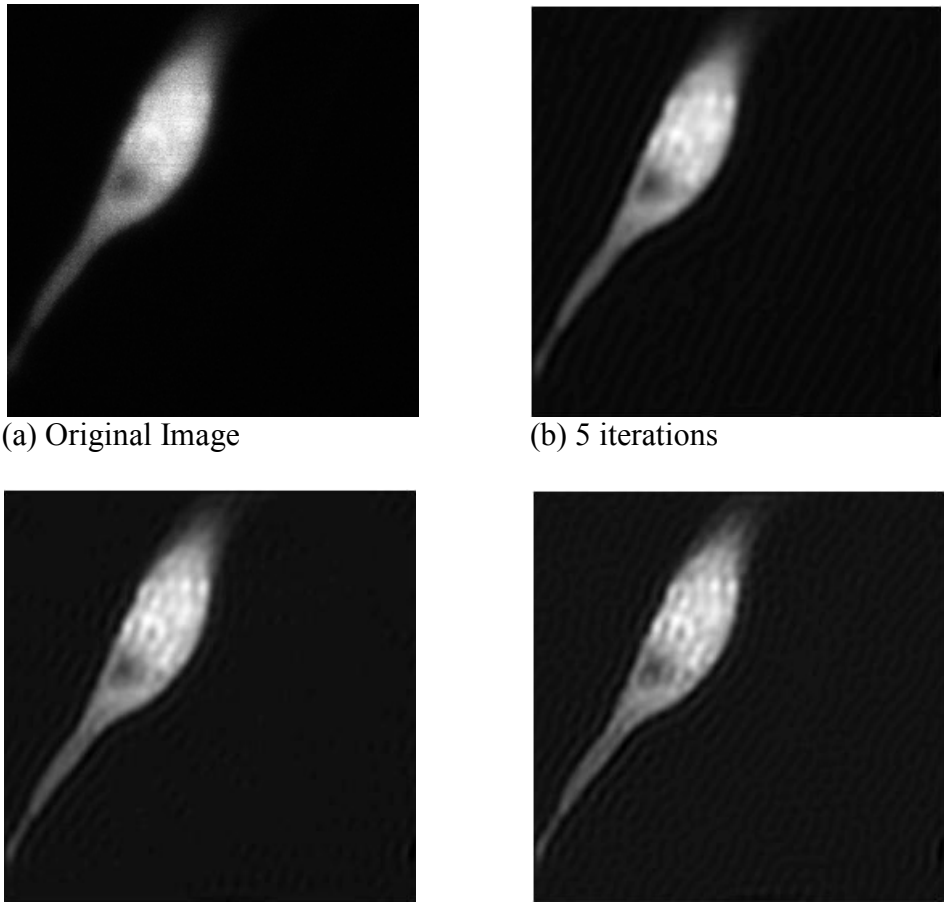


(b) *SD* = 4.5



(c) *SD* = 2.8

Figure 5.9: Deconvolution results on the test image for different blur estimates



(a) Original Image (b) 5 iterations

(c) 100 iterations (d) 1000 iterations

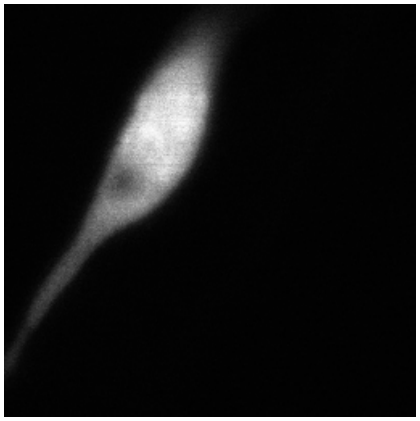
Figure 5.10: Iterative Wiener Filtering with Additive Correction estimates

Although the results of the Fourier-Wavelet Cascaded deconvolution process was not satisfactory for the training set image, we nonetheless explore its effectiveness in deconvolving this test image and obtaining a robust visual estimate. Figure 5.11 contains the results of applying this deconvolution method on the test image. Figure 5.11(e) contains a close-up version of the estimate obtained by Fourier-Wavelet Cascaded Deconvolution on the original test image. In order to emphasize the efficiency of the deconvolution algorithm to unveil extra information that was contained in the image itself, we invert the color scheme and observe the result in Figure 5.11(f). The cells inside can be made out easily because of their darker intensity against the background. Figure 5.12 shows the results of applying the Fourier-Wavelet Cascaded deconvolution algorithm on a bacteria image. The deconvolved estimate for the original image shown in Figure 5.12(a) can be observed in Figure 5.12(b) and is crisper than the original.

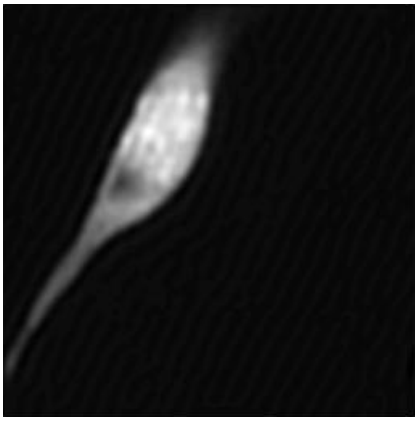
5.4 Conclusion

The objective of this research is to utilize signal processing algorithms to obtain a crisp version of a given degraded image without introducing any artifacts. Specifically, this being a blind deconvolution problem, we had to first compute the blur and noise in the given degraded image and utilize these parameters in the deconvolution process. The first part of this research work comprised of estimating the blur in the degraded image.

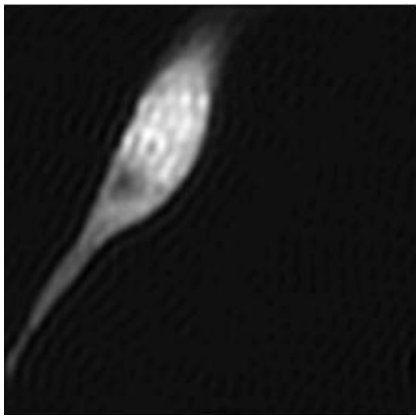
In Chapter 2, we propose an efficient Wavelet based algorithm that helps compute the Lipschitz exponent for high-frequency components in the image across wavelet scales. The motivation for this approach stems from Mallat's theorem that relates the Lipschitz exponent – a measure of the local regularity of the signal, to the variation of the modulus maxima of wavelet coefficients across scales. We use the Least Squares Fitting algorithm to compute Lipschitz exponent for each trace of wavelet coefficients across scales.



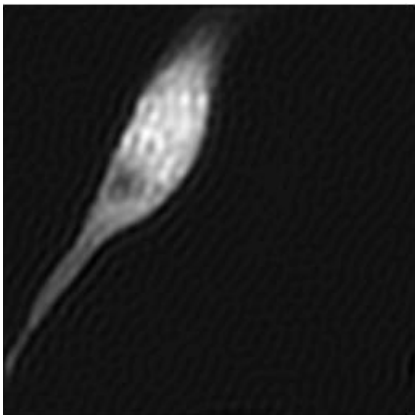
(a) Original Image



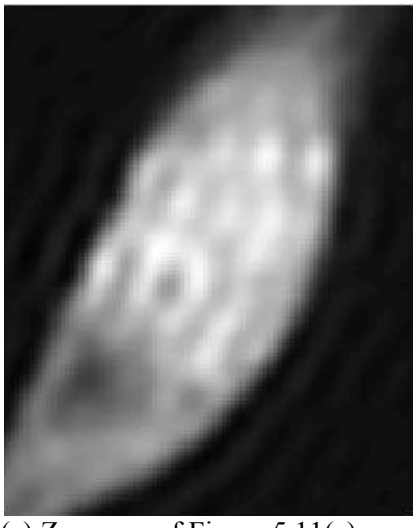
(b) 5 iterations



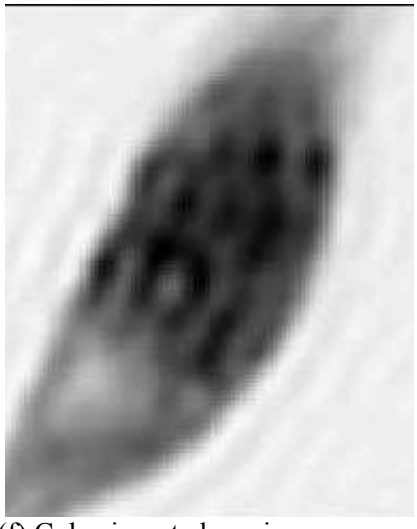
(c) 100 iterations



(d) 1000 iterations

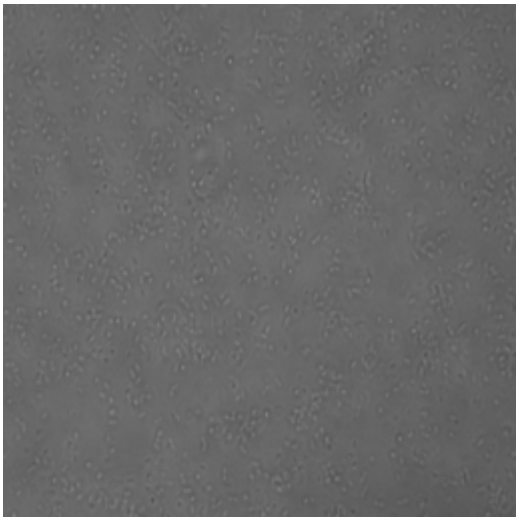


(e) Zoom-up of Figure 5.11(c)

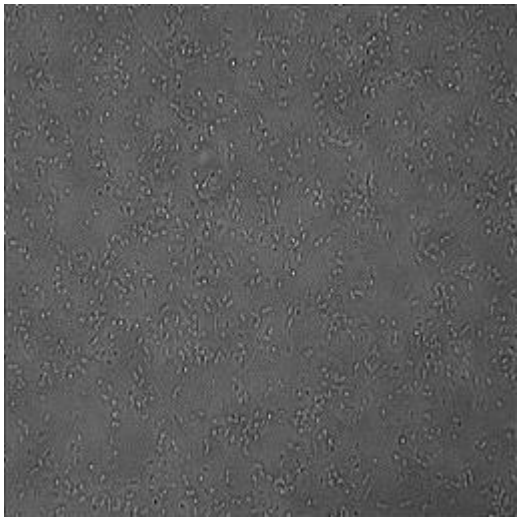


(f) Color-inverted version

Figure 5.11: Results of Fourier-Wavelet Cascaded Deconvolution process.



(a) Original Bacteria image



(b) After Cascaded Deconvolution

(c) Difference image

Figure 5.12. Deconvolved estimate for Bacteria image

In this work, we use a set of training images, which have very negligible blur, to obtain constants for relating the Mean Lipschitz exponent for each image to the standard deviation of the blur in the image. The computed estimates of standard deviation fall within 10% of the actual value of standard deviation of the applied blur. Using the fit values computed and applying it for the test image helps obtain an estimate of the standard deviation of the Gaussian blur for the test image.

In Chapter 3, we propose an efficient Regularized Iterative Wiener Filtering algorithm in the Fourier domain. Past methods have not fully exploited the convergence property of the Iterative Wiener Filtering process, especially, with the Additive Correction factor. Using the value of blur computed earlier, we perform Fourier domain shrinkage and the results are satisfactory – both on the training and test set images, as can be seen in Chapter 5. In Chapter 4, we define a Fourier-Wavelet Cascaded Deconvolution algorithm, which in essence, is basically adding a wavelet shrinkage block that operates on the Fourier domain estimate. The motivation for this cascaded approach comes from the fact that although Fourier domain represents noise economically, its representation of the edges and other high-frequency information in the image is not satisfactory. Hence, we obtain the noise-attenuated Fourier domain estimate and apply Wavelet domain shrinkage on this, given the fact that the wavelet domain provides an economical representation of singularities.

The Fourier domain estimate is visually far superior to the Fourier-Wavelet Cascaded estimate for the training set images. In the case of the test set image, the Fourier domain estimate again seems satisfactory, in that an appreciable amount of blur seems to have been done away with. But the Fourier-Wavelet Cascaded approach outperforms the Fourier domain estimate in the case of the test set image, in terms of visual quality. It can be concluded that the Fourier domain approach seems to provide a stable and satisfactory solution while the Fourier-Wavelet Cascaded approach seems image-dependent.

The estimation of blur in this experiment comprises of mainly estimating only the standard deviation, assuming the blur would be Gaussian. Improvements can be made in this aspect, of finding metrics to compute blur, in cases when it is not Gaussian. Also, use of features that complement local regularity analysis, would help improve the efficiency of the estimate. On the Deconvolution aspect, the issue of stability when using a Fourier-Wavelet Cascaded approach needs to be explored, as the results of this section are superior to the Fourier domain Filtering estimates, when they are bounded. Although Iterative Wiener Filtering with Additive Correction has been theoretically proven to achieve mathematical convergence, its results are just a small improvement on the more trivial Iterative Wiener Filtering approach and the results actually deteriorate with the number of iterations, in some cases. A better method to reach convergence would prove useful in delivering more robust results.

Bibliography

[1] S.Mallat and Wen Liang Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. Information Theory*, vol. 38, no. 8, pp. 617-643, 1992.

[2] S.Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[3] I.Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Inform. Theory*, vol. 36, pp. 961-1005, Sept. 1990.

[4] D.L.Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," Appl. Comput. Harmon. Anal., vol. 1, no. 1, pp. 100-115, Dec. 1993.

[5] Wolfram Research, http://mathworld.wolfram.com.

[6] A.Grossmann, "Wavelet transform and edge detection," *Stochastic Processes in Physics and Engineering*, M.Hazewinkel, Ed. Dodrecht: Reidel, 1986

[7] D.L.Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 613-627, May 1995.

[8] G. Davis and A. Nosratinia, "Wavelet-based image coding: An overview," in *Appl. Comput. Control Signals Circuits* (B. N. Datta, ed.), vol. 1, Birkhauser, 1999.

[9] A.D.Hillery and R.T.Chin, "Iterative Wiener filters for image restoration," *IEEE Trans. Signal Processing*, vol. 39, pp 1892-1899, Aug. 1991

[10] A.N.Tikhonov and V.Y.Arsenin, *Solutions of Ill-Posed Problems*. Washington D.C.: V.H.Winston & Sons, 1977.

[11] R.Neelamani, H.Choi, and R.G.Baraniuk, "ForWaRD: Fourier-Wavelet Regularized Deconvolution for Ill-Conditioned Systems," *IEEE Trans. Signal Processing*, Vol. 10, No. 20, 2003.

[12] D. L. Donoho, "Unconditional bases are optimal bases for data compression and for statistical estimation," *Appl. Comput. Harmon. Anal.*, vol. 1, no. 1, pp. 100–115, Dec. 1993.

[13] W. T. Freeman and E. H. Adelson, "The design and use of steerable Filters," *IEEE Trans. Pattern Analysis and Machine Intelligence* 13(9), pp. 891-906, 1991.

[14] D. L. Donoho and I. M. Johnstone, "Asymptotic minimaxity of wavelet estimators with sampled data," *Statist. Sinica*, vol. 9, no. 1, pp. 1–32, 1999.

[15] J.L.Starck, F.Murtagh, "Automatic noise estimation from the multiresolution support," *Publications Astron. Soc. Pacific* 110 (744)(1988) 193-199

[16] D. L. Donoho and I. Johnstone, "Minimax estimation by wavelet shrinkage," *Ann. Statist.*, vol. 26, pp. 879–921, 1998.

[17] J.L.Starck, F.Murtagh, "Multiscale entropy filtering," *ElSevier Signal Processing* 76 (1999) 147-165.

[18] S.Ghael, A.M.Sayeed and R.G.Baraniuk, "Improved wavelet denoising via empirical Wiener filtering," in *Wavelet Applications in Signal and Image Processing V*, Proc. SPIE, vol. 3169, Oct. 1997, pp.389-399.

[19] F.Rooms, M.Ronsse, A.Pizurica and W.Philips, "PSF Estimation with Applications in Auto Focus and Image Restoration," in *Proc. IEEE Benelux Signal Processing Symposium (SPS-2002),* Leuven, Belgium, March 21–22, 2002

[20] James Elder and Steven Zucker, “Local scale control for edge detection and blur estimation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 7, pp. 699–716, 1998.

[21] V. Kayargadde and J.-B. Martens, “Estimation of edge parameters and image blur from local derivatives,” *Journal on Communications*, pp. 33–34, 1994.

[22] F.Rooms, W.Philips and P.V.Oostveldt, “Estimation of Anisotropic Blur for the Restoration of confocal images,” SPIE 2003.

Vita

Ravi Viswanathan received the B.E degree in 2002 from Bharathidasan University, Tiruchirappalli, India, in Electrical and Electronics Engineering. After working as a hardware design engineer at SCM Microsystems in Chennai, India for a year, he joined the M.S program at the University of Tennessee in 2003 with a major in Engineering Sciences. His research interests include Image Processing, Pattern Recognition and Artificial Neural Networks.

Mr.Viswanathan won the first prize in all-India student paper contests in 2000 and 2002 in Electrical Engineering and was consistently ranked within the top 2% of his peers throughout his Bachelors.