DEVELOPMENT OF IMAGE RESTORATION TECHNIQUES

THESIS REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

Computer Science and Engineering

by

TONY XAVIER

Department of Computer Science and Engineering

National Institute of Technology

Rourkela

May 2007

DEVELOPMENT OF IMAGE RESTORATION TECHNIQUES

THESIS REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology

in

Computer Science and Engineering

by

TONY XAVIER

Under the Guidance of

Prof.BANSHIDHAR MAJHI

Department of Computer Science and Engineering

National Institute of Technology

Rourkela

May 2007

National Institute of Technology

Rourkela

CERTIFICATE

This is to certify that the thesis entitled,*Development of Image Restoration Techniques* submitted by Sri.**Tony Xavier** in partial fulfillment of the requirements for the award of Master of Technology Degree in Computer Science and Engineering at National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University / Institute for the award of any Degree or Diploma.

Prof. Banshidhar Majhi

Professor

Dept. of Computer Science and Engg.

National Institute of Technology

Rourkela-769008

21 May 2007

Acknowledgments

There are a few people without whose help this thesis would have been incomplete. First among them is of course Prof.Banshidhar Majhi for his excellent guidance and motivation he offered from the beginning to the end of the thesis. Next, my gratitude goes to Pankaj Kumar Sa, Lecturer of our department, for his guidance and help whenever I required. I am thankful to Prof. S K Jena, Head of our department for providing with all the facilities required. Also, I am thankful to all the Professors of our department for their help and encouragement they offered all the time. Last but not the least is my friends who helped throughout the year to get over all difficulties no matter technical or personal.

TONY XAVIER

Contents

ABSTRACT

Image denoising and image deblurring are studied as part of the thesis. In deblurring, blind deconvolution is investigated. Out of the several classes of blind deconvolution techniques, Non parametric Methods based on Image Constraints are studied at greater depth. A new algorithm based on the Iterative Blind Deconvolution(IBD) technique is developed. The algorithm makes use of spatial domain constraints of non-negativity and support. The Fourier-domain constraint may be described as constraining the product of the Fourier spectra of the image f and the Fourier spectra of the point spread function h to be equal to the convolution spectrum. Within each iteration, the algorithm switches between spatial domain and frequency domain and imposes known constraints on each.

The convergence of the original IBD can be accelerated by limiting high magnitude values in frequency domain of both estimated image and point spread function. The new algorithm converges within less than 25 iterations where as the original IBD took nearly 500 iterations. Inclusion of the support constraint in the spatial domain improves quality of the restored image. Also, sum of the spatial domain values of the point spread function should be made equal to one at the end of each iteration, for preventing the loss of image intensity. PSNR values calculated for restored images show significant improvement in image quality. A PSNR of 17.8dB is obtained for the improved scheme where as it is 14.3dB for the original IBD. A new stopping criteria based on standard deviation of the image power for last k iterations is defined for stopping the algorithm when it converges.

In denoising, an edge retrieval technique is developed which preserves the image details along with effectively removing impulse noise. Noisy pixels are detected in the first phase and in the next phase those pixel values are replaced with an estimate of the actual value. For dealing with the wrong classification of edge pixels as noisy pixels, an *edge retrieval technique* based on pixel-wise MAD is defined. This scheme retrieves the pixels which are wrongly classified as noise. The algorithm gives high PSNR values as well as very good detail preservation.

List of Figures

Chapter 1

INTRODUCTION

Image Restoration is the process of reconstructing or recovering an image that has been degraded by some degradation phenomenon. Restoration techniques are primarily modeling of the degradation and applying the inverse process in order to recover the original image. Image restoration techniques exist both in spatial and frequency domain.

1.1 A MODEL OF IMAGE DEGRADATION

An image may be described as a two-dimensional function I

$$I = f(x, y) \tag{1.1}$$

where x and y are spatial coordinates. Amplitude of f at any pair of coordinates (x, y) is called intensity I or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called digital image.If $f(x, y)$ is the original image, $h(x, y)$ is a degradation function and $\eta(x, y)$ is the additive noise then the degraded image $g(x, y)$ is given as [9]:

$$g(x, y) = f(x, y) * h(x, y) + \eta(x, y) \tag{1.2}$$

where the symbol $*$ indicates spatial convolution. Since convolution in spatial domain is equal to multiplication in the frequency domain, the corresponding frequency domain representation is given as:

$$G(u, v) = F(u, v)H(u, v) + N(u, v) \tag{1.3}$$

where the terms in capital letters are the Fourier transforms of the corresponding terms in equation(1.2)

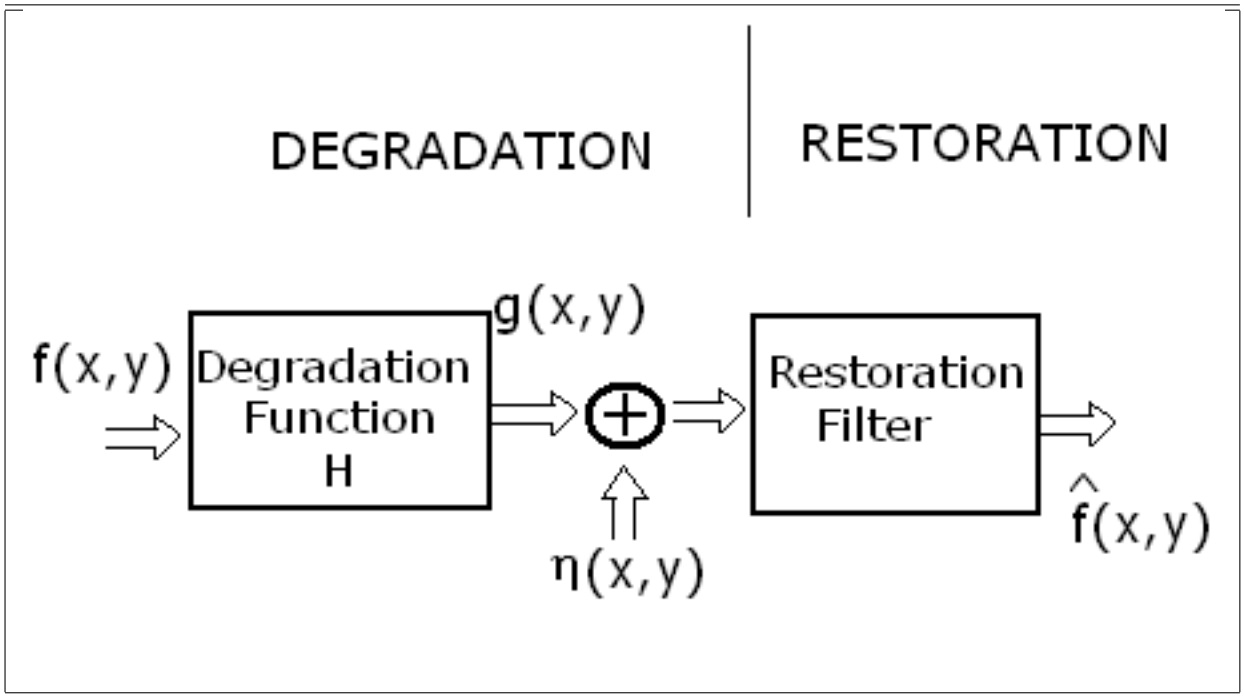

Figure 1.1: A model of the image degradation/restoration process.

Many types of the degradations can be approximated by linear, position invariant processes. Since degradations are modeled as being the result of convolution, and restoration seeks to find filters that apply the process in reverse, the term *image deconvolution* is used to signify linear image restoration.

1.2 POINT SPREAD FUNCTION

The linear position-invariant function $h(x, y)$ in equation(1.2) is known as a point spread function. The point spread function gets convolved with the original image to give the degraded image. Some commonly occurring image degradations, which are linear and position-invariant are given below.

Motion Blur

We often see images blurred because of camera movement during image capture. Suppose the relative motion is of velocity ν at an angle θ with the horizontal axis and if T is the duration of exposure, then the blur length is $L = \nu T$ and the motion blur PSF can be expressed as

$$h(x,y) = \begin{cases} 1/L & if \quad 0 \leq |x| \leq L\cos\theta; \quad y = Lsin\theta \\ 0 & otherwise \end{cases} \tag{1.4}$$

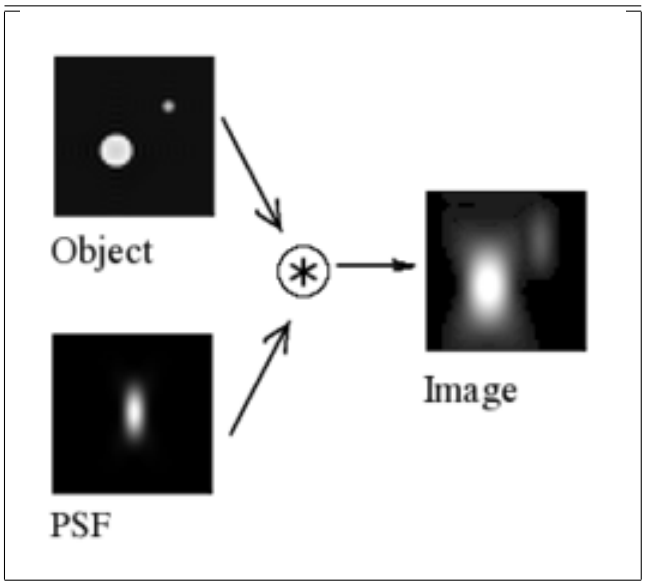

Figure 1.2: Convolution of the point spread function with a point object gives the observed image.

Camera Defocus

Another commonly occurring blur is because of improperly focussed camera. Assuming the lens system is of circular aperture, with radius r the point spread function can be expressed as

$$h(x,y) = \begin{cases} 0 & if \quad \sqrt{x^2+y^2} > r \\ 1/\pi r^2 & otherwise \end{cases} \tag{1.5}$$

1.3 DECONVOLUTION

If we can estimate the point spread function, $h(x,y)$ which caused the degradation, then we can get back the original image by deconvolution. There are two classes of deconvolution techniques. Classical image restoration techniques and blind deconvolution techniques. In classical restoration, we need to have prior knowledge of the PSF which caused the degradation. Blind deconvolution techniques are used when we don't have any prior knowledge of the degradation process.

1.3.1 Classical Restoration Techniques

Classical restoration techniques need prior knowledge of the degradation process. The degradation can be estimated by one of the several techniques [9] given below.

- Estimation by image observation.

- Estimation by experimentation
- Estimation by modeling

After estimating the point spread function, one of the three techniques [9] given below can be used for deconvolution.

Inverse Filtering

Direct inverse filtering is the simplest approach to restoration. In this method, an estimate of the Fourier transform of the image $\hat{F}(u,v)$ is computed by dividing the Fourier transform of the degraded image by the Fourier transform of the degradation function.

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \tag{1.6}$$

This method works well when there is no additive noise in the degraded image. That is, when the degraded image is given by $g(x,y) = f(x,y) * h(x,y)$. But if noise gets added to the degraded image then the result of direct inverse filtering is very poor. Equation 1.3 gives the expression for $G(u,v)$. Substituting for $G(u,v)$ in the above equation, we get

$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)} \tag{1.7}$$

The above equation shows that direct inverse filtering fails when additive noise is present in the degraded image. Because noise is random and so we can not find the noise spectrum $N(u,v)$.

Minimum Mean Square Error Filtering

Minimum mean square filtering, also known as Wiener Filtering, is more robust in the presence of additive noise. Wiener filtering incorporates both degradation function and statistical characteristics of noise into the restoration process. The objective of this technique is to find an estimate $\hat{f}$ of the original image f such that the mean square error between them is minimized. This error measure is given by

$$e^2 = E\left\{(f - \hat{f})^2\right\} \tag{1.8}$$

where $E\{.\}$ is the expected value of the argument. The minimum of the error function in the above equation is given in the frequency domain by the expression

Figure 1.3: (a)Lena image of size 256×256. (b) Lena image blurred with motion blur of length 31 pixels and angle 45 degree.

Figure 1.4: (a)Motion blurred Lena image in Fig.(1.3b) restored using direct inverse filtering. (b) Image restored using direct inverse filtering when gaussian noise with mean zero and variance 1 is added to the image.

$$\begin{aligned}
\hat{F}(u,v) &= \left[\frac{H^*(u,v)S_f(u,v)}{S_f(u,v)|H(u,v)|^2 + S_\eta(u,v)}\right] G(u,v) \\
&= \left[\frac{H^*(u,v)}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v) \\
&= \left[\frac{1}{H(u,v)}\frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v) \qquad (1.9)
\end{aligned}$$

where $H^*(u,v)$ is the complex conjugate of $H(u,v)$ and $|H(u,v)|^2 = H^*(u,v)H(u,v)$

$S_\eta(u,v) = |N(u,v)|^2$ = power spectrum of the noise

$S_f(u,v) = |F(u,v)|^2$ =power spectrum of the original image.

If the noise is zero, then the noise power spectrum vanishes and the Wiener filter reduces to the inverse filter. But since it is not possible to get the power of the original image, power spectrum of the degraded image can be used.

Figure 1.5: (a)Blurred image of figure(1.3b) deconvolved using weiner filter. (b)Blurred image of figure(1.3b) deconvolved using constrained least square filtering.

Constrained Least Squares Filtering

The method of using a constant for the ratio of power spectra is not a suitable solution always. Constrained least square filtering require knowledge of only the mean and variance of the noise. The method finds the minimum of a criterion function C, defined as

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\nabla^2 f(x,y)\right]^2 \tag{1.10}$$

subject to the constraint

$$\| g - H\hat{f} \|^2 = \| \eta \|^2 \tag{1.11}$$

where $\| \cdot \|$ is the Euclidean vector norm and $\hat{f}$ is the estimate of the original image. The laplacian operator ∇^2 is defined as

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \tag{1.12}$$

The frequency domain solution for this optimization problem is given by the expression

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma|P(u,v)|^2}\right] G(u,v) \tag{1.13}$$

where γ is a parameter that must be adjusted so that the constraint mentioned above is satisfied, and $P(u,v)$ is the Fourier transform of the function

$$p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{1.14}$$

which is actually the Laplacian operator.

1.3.2 Blind Deconvolution

All the techniques described so far, requires the knowledge of the exact degradation function. Then the restoration algorithm is applied to invert the degradation process. Those techniques are called the Classical Restoration Techniques. If the deconvolution is performed without having prior knowledge of the degradation function, then it is called *blind deconvolution* [12]. In blind deconvolution, both the degradation function and the true image are estimated from the degraded image characteristics. Partial information about the imaging system may also be utilized if available.

Blind deconvolution is of great interest because in most of the practical cases, knowing the degradation function is not possible. For example, in applications like remote sensing and astronomy, it is difficult to statistically model the original image or even know specific information about scenes never imaged before.

1.4 DENOISING

Another type of image degradation is due to additive noise. Random values get added to the intensity values. Denoising involves the application of some filtering technique to get the true image back. The denoising algorithms [20] vary greatly depending on the type of noise present in the image. Each type of image is characterized by a unique noise model. Each noise model corresponds to a probability density function which describes the distribution of noise within the image.

1.4.1 Noise Models

The spatial component of noise is based on the statistical behaviour of the intensity values. These may be considered as random variables , characterized by a probability density function (PDF). Some commonly found noise models [9] and their corresponding PDFs are given below.

Gaussian Noise

Gaussian noise is noise that has a probability density function of the normal distribution (also known as Gaussian distribution). In other words, the values that the noise can take on are Gaussian distributed. If z is a Gaussian random variable representing noise, then its PDF is given by

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \tag{1.15}$$

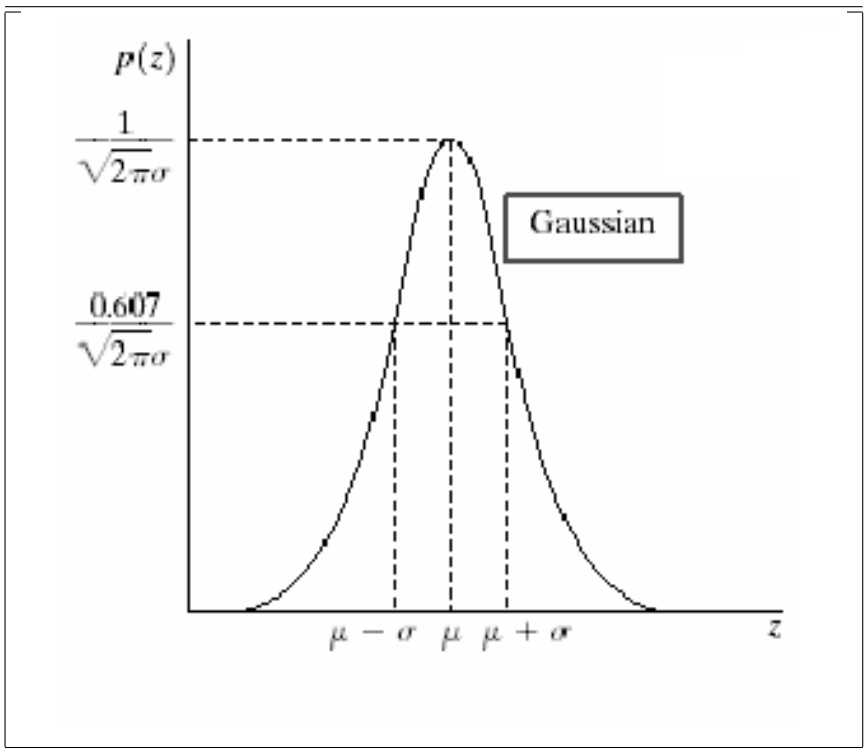

Figure 1.6: The probability density function for Gaussian Distribution

Gaussian noise is additive noise. That is the Gaussian distributed noise values get added to the intensity values of the image.

Uniform Noise

Another commonly observed image noise is uniform noise. In this case the noise can take on values in an interval $[a, b]$ with uniform probability. The PDF of uniform noise is given by

$$p(z) = \begin{cases} \frac{1}{b-a} & if a \leq z \leq b \\ \\ 0 & otherwise \end{cases} \tag{1.16}$$

The plot of the probability density function for uniform noise distribution is given in figure(1.7a).

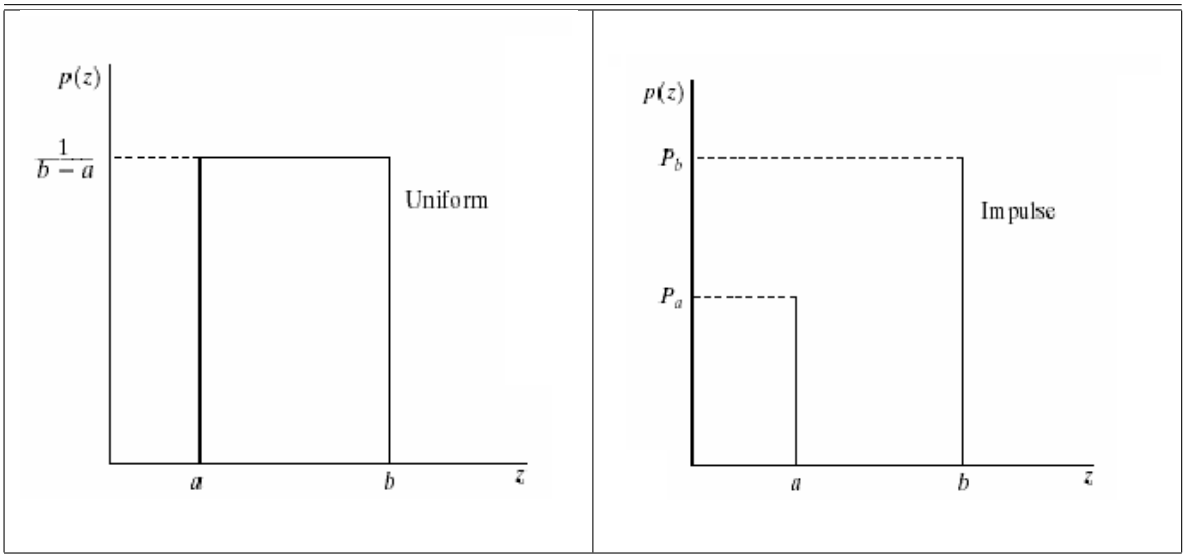

Figure 1.7: (a)Probability density function for Uniform Noise. (b) Probability density function for salt and pepper noise

Impulse Noise

Impulse noise is characterized by a noise spike replacing the actual pixel value. Impulse noise is further divided into two classes. *Random valued impulse noise*(RVIN) and *salt & pepper noise*(SPN). In RVIN, the impulse value at a particular pixel may be a random value between a particular interval. But in SPN the impulses are either the minimum value or the maximum value allowed in the intensity values. For example 0 and 255 in the case of 8-bit image.

1.4.2 Denoising Techniques

When the only degradation present in an image is noise, then equation(1.2) becomes

$$g(x, y) = f(x, y) + \eta(x, y) \tag{1.17}$$

and Eq(1.3) becomes

$$G(u, v) = F(u, v) + N(u, v) \tag{1.18}$$

Denoising techniques exist in both spatial domain as well as frequency domain.

Spatial Filtering

Spatial filtering is preferred when only additive noise is present. The different classes [9] of filtering techniques exist in spatial domain filtering.

- Mean Filters
- Order-Statistics Filters
- Adaptive Filters

Arithmetic mean filter

This belongs to the category of mean filters. In this method the middle pixel value of the filter window is replaced with the arithmetic mean of all the pixel values within the filter window. A mean filter simply smoothes local variations in an image. Noise is reduced as a result of this smoothening, but edges within the image get blurred.

Median Filter

Median filter comes under the class of order-statistics filters. Response of Order-statistics filters is based on ordering the pixels contained in the filter window. Median filter replaces the value of a pixel by the median of the gray levels within the filter window. Median filters are particularly effective in the presence of impulse noise.

Adaptive Filters

Adaptive filters change its behaviour based on the statistical characteristics of the image inside the filter window. Adaptive filter performance is usually superior to non-adaptive counterparts. But the improved performance is at the cost of added filter complexity .

Mean and variance are two important statistical measures using which adaptive filters can be designed. For example if the local variance is high compared to the overall image variance, the filter should return a value close to the present value. Because high variance is usually associated with edges and edges should be preserved.

Chapter 2

BLIND DECONVOLUTION

In the first chapter we have seen that if we can estimate the point spread function, $h(x, y)$ which caused the degradation, then we can get back the actual original image by deconvolution. But unfortunately, in many practical situations, the blur is often unknown, and little information is available about the true image. Therefore, the true image $f(x, y)$ must be identified directly from $g(x, y)$ by using partial or no information about the blurring process and the true image. Such an estimation problem, assuming the linear degradation model of equation(1.2) is called blind deconvolution.

There are several motivating factors behind the use of blind deconvolution for image processing applications. In practice, it is often costly, dangerous, or physically impossible to obtain a priori information about the imaged scene. For example, in applications like remote sensing and astronomy, it is difficult to statistically model the original image or even know specific information about scenes never imaged before . In addition, the degradation from blurring cannot be accurately specified. In aerial imaging and astronomy, the blurring cannot be accurately modeled as a random process, since fluctuations in the PSF are difficult to characterize. In real-time image processing, such as medical video-conferencing, the parameters of the PSF cannot be pre-determined to instantaneously deconvolve images. Moreover, on-line identification techniques used to estimate the degradation may result in significant error, which makes the restored image useless.

In other applications, the physical requirements for improved image quality are un realizable. For instance, in space exploration, the physical weight of a high resolution camera exceeds practical constraints. Similarly, in x-ray imaging, improved image qual-

ity occurs with increased incident x-ray beam intensity, which is hazardous to a patient's health. Thus, blurring is unavoidable. In such situations, the hardware available to measure the PSF of an imaging system is often difficult to use. Although these methods work well to identify the PSF, they are esoteric, which limits their wide use. Blind deconvolution is a viable alternative.

2.1 APPROACHES IN BLIND DECONVOLUTION

There are two main approaches to blind deconvolution of images:
1. Identifying the PSF separately from the true image, in order to use it later with one of the known classical image restoration methods. Estimating the PSF and the true image are disjoint procedures. This approach leads to computationally simple algorithms.
2. Incorporating the identification procedure with the restoration algorithm. This merge involves simultaneously estimating the PSF and the true image, which leads to the development of more complex algorithms.

2.1.1 Classification of Blind Techniques

Blind deconvolution techniques can be further divided into five broad categories as shown in figure(2.1). Each of those categories contain several algorithms for blind deconvolution. Some of the algorithms are in frequency domain and some are in spatial domain. A few algorithms make use of both frequency domain and spatial domain data. The five classes of algorithms are given below. A detailed classification of blind deconvolution techniques can be found in [12].

- A priori Blur Identification Methods
- Zero Sheet Separation
- ARMA Parameter Estimation Methods
- Non parametric Methods Based on Image Constraints
- Non Parametric methods Based on Higher Order Statistics

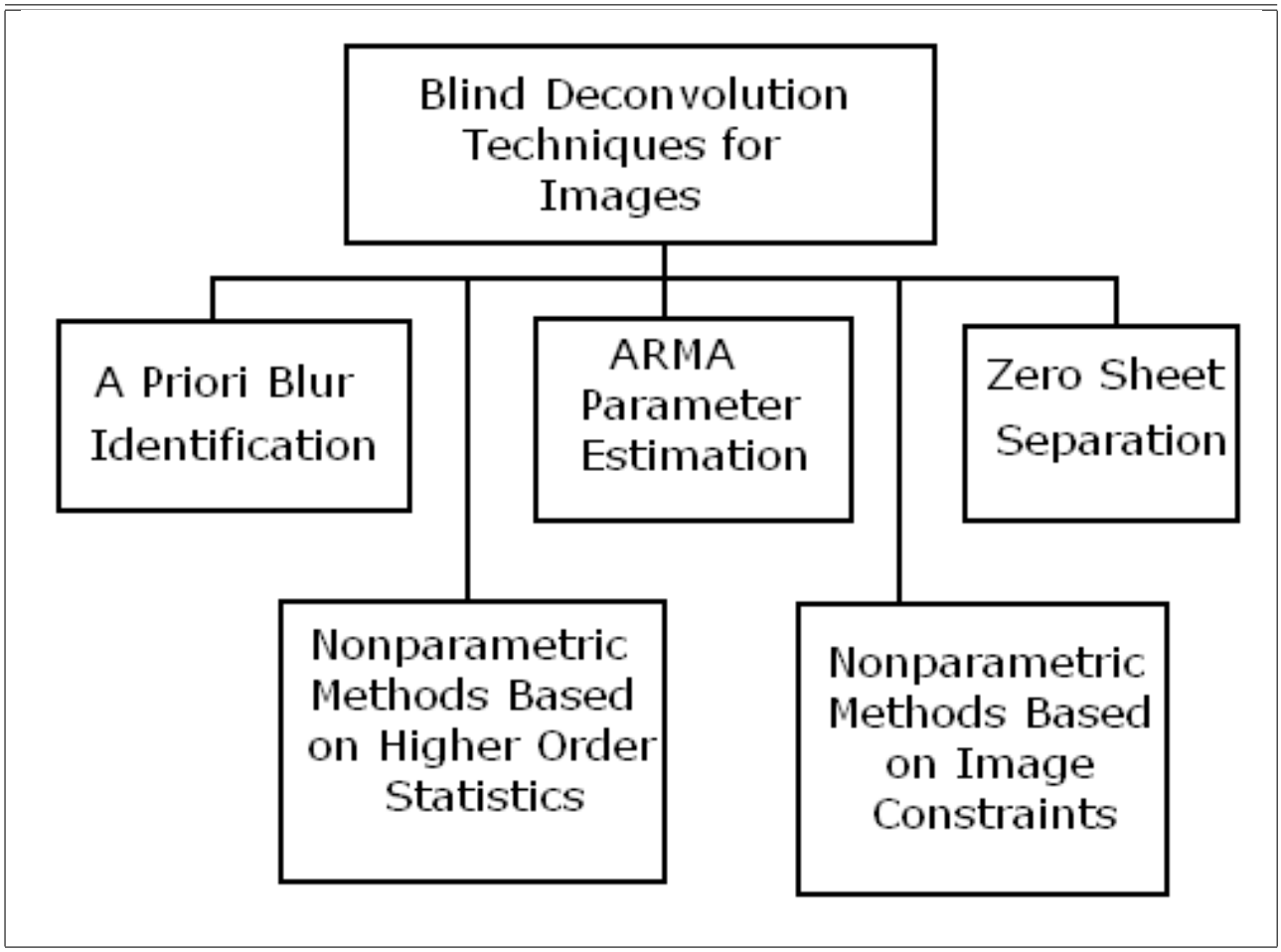

Figure 2.1: The classification of blind deconvolution techniques

2.2 A PRIORI BLUR IDENTIFICATION

A priori blur identification methods perform blind deconvolution by identifying the PSF prior to restoration. This general class of techniques makes assumptions on the characteristics of the PSF such as symmetry, and availability of a known parametric form of the blur. Popular parametric models include PSFs resulting from linear camera motion or an out-of-focus lens system. Based on these assumptions, an attempt is made to completely characterize the PSF using special features of the true/blurred image. Once the PSF has been completely identified, one of the classical restoration techniques is used to estimate the true image using deconvolution.

A priori blur identification techniques are the simplest class of blind deconvolution methods to implement and have low computational requirements. They are applicable to situations in which the true image is known to possess special features, or when the PSF is known to be of a special parametric form. For more general situations or when

less information is available other deconvolution algorithms must be used.

2.2.1 Cepstrum Based Motion Blur Identification

A method for identifying linear motion blur is to compute the two-dimensional cepstrum of the blurred image $g(x, y)$ [11]. The cepstrum of $g(x, y)$ is given by

$$\mathcal{C}(g(x, y)) = \mathcal{F}^{-1}(log(|\mathcal{F}(g(x, y))|). \tag{2.1}$$

An important property of the cepstrum is that it is additive under convolution. Thus, ignoring noise, we have

$$\mathcal{C}(g(x, y)) = \mathcal{C}(f(x, y)) + \mathcal{C}(h(x, y)) \tag{2.2}$$

$\mathcal{C}(h(x, y)) = \mathcal{F}^{-1}(log(|\mathcal{F}(h(x, y))|)$ has large negative spikes at a distance L from the origin. By the additivity of the cepstrum, this negative peak is preserved in $\mathcal{C}(g(x, y))$, also at a distance L from the origin. If the noise level of the blurred image is not too high, there will be two pronounced peaks in the cepstrum. To estimate the angle of motion blur, draw a straight line from the origin to the first negative peak. The angle of motion blur is approximated by the inverse tangent of the slope of this line.

Figure 2.2: (a)The original image. (b) The original image blurred with a motion blur of length 31pixels and angle 0^o. (c) Image restored using cepstral method.

Limitations of the Cepstrum Based Technique

From figure(2.2), it is evident that the cepstrum based method gives excellent results for horizontal motion blur. Similar results were obtained for vertical motion blur also. The

restoration is perfect since the length estimation was accurate. But the length estimate shows significant error when the motion blur direction is not horizontal or vertical.

Chapter 3

NON PARAMETRIC TECHNIQUES BASED ON IMAGE CONSTRAINTS

The algorithms of this class do not assume parametric models for either the image or the blur. Deterministic constraints such as non negativity, known finite support, and existence of blur invariant edges are assumed for the true image. A number of blind deconvolution techniques for images fall into this class, which include the Iterative Blind Deconvolution algorithm , Simulated Annealing algorithm , Non negativity And support constraints Recursive Inverse Filtering (NAS-RIF)algorithm.
The methods are iterative and simultaneously estimate the pixels of the true image and the PSF (or its inverse). The constraints on the true image and the PSF are incorporated into an optimality criterion which is minimized using numerical techniques.

3.1 DETERMINISTIC IMAGE CONSTRAINTS

The deterministic constraints used in this particular class of techniques include non negativity and known finite support. Non negativity can be assumed for both image pixel values as well as PSF coefficients. Another constraint that can be used is a known finite support. Support is the smallest rectangle which known to encompass the true image. Blurring causes the image to spread outside the support. Restoration algorithm may make use of the fact that true image pixels can not lie outside the support. The concept

of support is illustrated in figure(3.1). Another spatial domain constraint which is found to be useful is the fact that the sum of the PSF coefficients will always be equal to one. This constrained may be enforced in each iteration of the algorithm.

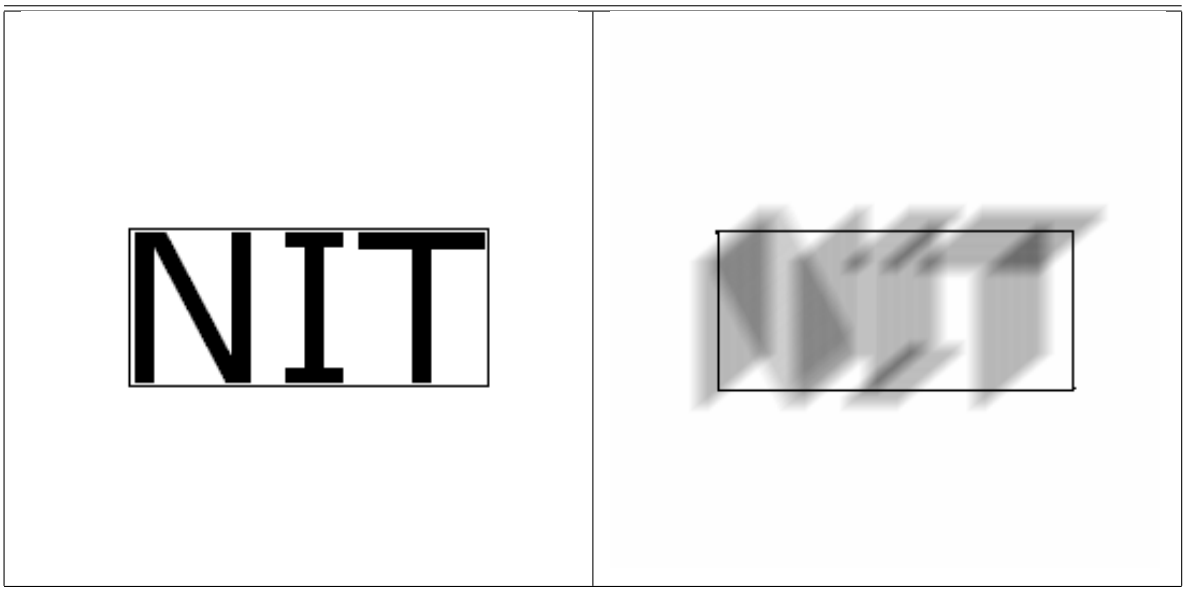

Figure 3.1: (a).An image with the support surrounding it. (b) The image gets spread outside the support after blurring.

3.2 NON NEGATIVITY AND SUPPORT CONSTRAINTS RECURSIVE INVERSE FILTERING

Non negativity and support constraints recursive inverse filtering (NAS-RIF) [13] technique makes the assumption of non negativity on the true image. That is, the algorithm assumes the true image pixel intensity values will be nonnegative. The knowledge of a known support is also assumed. The only assumptions made on the PSF, however, is that it is absolutely summable and that it has an inverse $h^{-1}(x, y)$ which is also absolutely summable. An advantage of this method is that it does not require the PSF to be of known finite extent, as do the other methods; this information is often difficult to obtain.

The NAS-RIF technique is shown in figure(3.2). It consists of a variable FIR filter $u(x, y)$ with the blurred image $g(x, y)$ as input. The output of this filter represents

an estimate of the true image $\hat{f}(x, y)$. This estimate is passed through a nonlinear filter, which uses a non-expansive mapping to project the estimated image into the space representing the known characteristics of the true image. The difference between the projected image $f_{NL}(x, y)$ and $f(x, y)$ is used as the error signal to update the variable filter u(x,y).

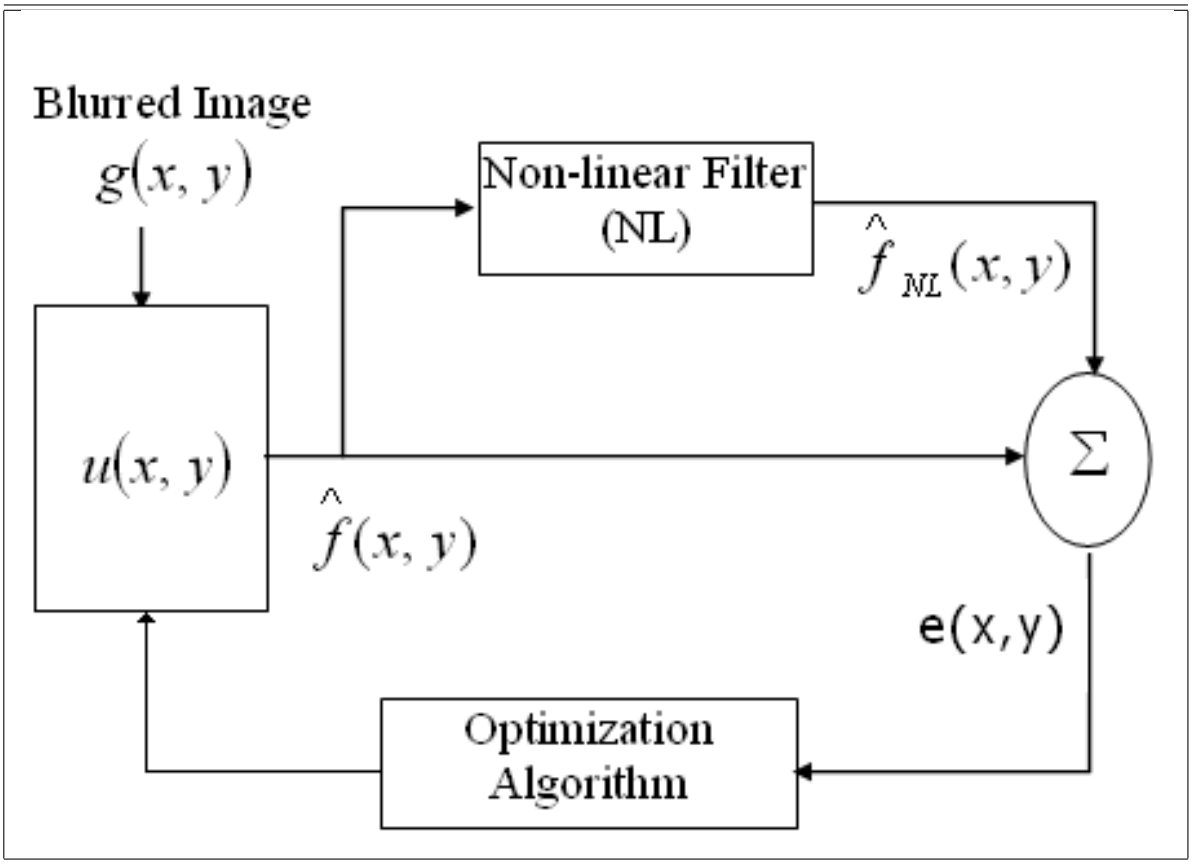

Figure 3.2: Block Diagram of the NASRIF algorithm

If we assume the image is nonnegative with known support, the NL block of Fig. 3.2 represents the projection of the estimated image onto the set of images that are nonnegative with given finite support. Thus, the negative pixel values within the region of support must be zero, and the pixel values outside the region of support are the background grey-level, L_B. The cost function used in the restoration procedure is defined as

$$J = \sum_{\forall(x,y)} \left[\hat{f}_{NL}(x, y) - \hat{f}(x, y)\right]^2 \tag{3.1}$$

where

$$\hat{f}_{NL}(x,y) = \begin{cases} \hat{f}(x,y) & if \quad \hat{f}(x,y) > 0 \quad and \quad (x,y) \in SUP \\ 0 & if \quad \hat{f} < 0 \quad and \quad (x,y) \in SUP \\ L_B & if \quad (x,y) \in \overline{SUP} \end{cases} \tag{3.2}$$

Where SUP is the region of the image within the support and $\overline{SUP}$ is the region outside the support. Substituting for $\hat{f}_{NL}$ in equation(3.1) gives

$$\begin{aligned} J &= \sum_{(x,y)\in SUP} \hat{f}^2(x,y) \left[\frac{1 - sgn(\hat{f}(x,y))}{2}\right] \\ &+ \sum_{(x,y)\in\overline{SUP}} \left[\hat{f}(x,y) - L_B\right]^2 \end{aligned} \tag{3.3}$$

where the definition for $sgn(.)$ is

$$sgn(f) = \begin{cases} 1 & if \quad f \geq 0 \\ -1 & if \quad f < 0 \end{cases} \tag{3.4}$$

The parameter set $u(x,y) = 0$ for all (x,y) globally minimizes J. This results in a restored image $\hat{f}(x,y) = 0$ for all (x,y) , which is the all black solution. To avoid this trivial solution, we make use of an assumption, that the sum of all the PSF coefficients is positive, i.e.,

$$\sum_{\forall(x,y)} h(x,y) > 0 \tag{3.5}$$

Also the sum of inverse filter coefficients is assumed to be 1, so that the filtering doesn't cause any loss of total image intensity. One option for constraining the parameters to fulfill this condition is to normalize $u(x,y)$ at every iteration. Another option is to use a penalty method and add a third term to the cost function. The overall function is then represented by

$$\begin{aligned} J &= \sum_{(x,y)\in SUP} \hat{f}^2(x,y) \left[\frac{1 - sgn(\hat{f}(x,y))}{2}\right] \\ &+ \sum_{(x,y)\in\overline{SUP}} \left[\hat{f}(x,y) - L_B\right]^2 + \gamma \left[\sum_{\forall(x,y)} u(x,y) - 1\right]^2 \end{aligned} \tag{3.6}$$

The cost function consists of three components. The first penalizes the negative pixels of the image estimate inside the region of support, and the second penalizes the pixels of

the image estimate outside the region of support that are not equal to the background color. The first component prevents the pixels of the intermediate restorations from becoming highly negative. The conjugate gradient minimization routine is used for the minimization of the cost function. NAS-RIF algorithm is summarized below.

- Set initial conditions:
  $u_0 = [u_0(1,1), \ldots, u_0((U_x+1)/2, (U_y+1)/2), \ldots, u_0(U_x, U_y)] = [0, \ldots, 1, \ldots, 0]$
  $tolerance: \quad \delta > 0$

- At iteration k,(k=1,2,3,...)

  1. If $J(u_k) \leq \delta$ then stop

  2. Calculate the gradient vector of J
     $f_k(x,y) = u_k(x,y) * g(x,y)$

$$\begin{aligned}
\left[\nabla J(u_k)_{j+(i-1)*U_x,1}\right] &= \frac{\partial J(u_k)}{\partial u(i,j)} \\
&= 2 \sum_{(x,y)\epsilon SUP} \hat{f}_k(x,y) cl(\hat{f}_k(x,y)) g(x-i+1, y-j+1) \\
&+ 2 \sum_{(x,y)\in \overline{SUP}} [f_k(x,y) - L_B]\, g(x-i+1, y-j+1) \\
&+ 2\gamma \left[\sum_{\forall(x,y)} u(x,y) - 1 \right] \qquad (3.7)
\end{aligned}$$

  3. If $k = 0, d_k = -\nabla J(u_k)$.
     Otherwise,
     $a)\, \beta_{k-1} = \frac{\langle \nabla J(u_k) - \nabla J(u_{k-1}), \nabla J(u_k) \rangle}{\|\nabla J(u_{k-1})\|^2}$

     $b)\, d_k = -\nabla J(u_k) + \beta_{k-1} d_{k-1}$

  4. Find t_k such that

$$J(u_k + t_k d_k) \leq J(u_k + t d_k) \quad for \quad t \in \mathbb{R}$$

  5. $u_{k+1} = u_k + t_k d_k$

3.2.1 Limitations of NAS-RIF

The NAS-RIF algorithm did not yield results as expected. NAS-RIF algorithm makes use of constraints only in spatial domain, not in spatial domain. That may be the reason for the unsatisfactory performance. Minimizing the cost function may result in one of several solutions, most of which are physically meaningless to the problem.

3.3 ITERATIVE BLIND DECONVOLUTION

The degraded image $g(x, y)$ is represented by the convolution of the original image $f(x, y)$ and the point spread function $h(x, y)$.

$$g(x, y) = f(x, y) * h(x, y) \tag{3.8}$$

The equivalent equation in frequency domain is given by

$$G(x, y) = F(x, y)H(x, y) \tag{3.9}$$

Iterative blind deconvolution technique [3] uses some general a priori information concerning the original image $f(x, y)$ and the PSF $h(x, y)$. After a random initial guess is made for the PSF, the algorithm alternates between the image and Fourier domains, enforcing known constraints in each. The constraints are based on information available about the image and PSF.

The basic deconvolution method consists of the following steps.
First, a nonnegative-valued initial estimate of the PSF $\hat{h}(x, y)$ is input into the iterative scheme. This function is Fourier transformed to yield $\hat{H}(u, v)$. which is then inverted to form an inverse filter and multiplied by $G(u, v)$ to form a first estimate of the original image spectrum $F(u, v)$. This estimated Fourier spectrum is inverse transformed to give $f(x, y)$. The image-domain constraint of non negativity is now imposed by putting to zero all pixels of the image $f(x, y)$ that have a negative value. A positive constrained estimate $\hat{f}(x, y)$ is formed that is Fourier transformed to give the spectrum $\hat{F}(u, v)$. This is inverted to form another inverse filter and multiplied by $G(u, v)$ to give the next spectrum estimate $H(u, v)$. A single iterative loop is completed by inverse Fourier transforming $H(u, v)$ to give $h(x, y)$ and by constraining this to be nonnegative, yielding the next function estimate $\hat{h}(x, y)$. The iterative loop is repeated until two positive functions with the

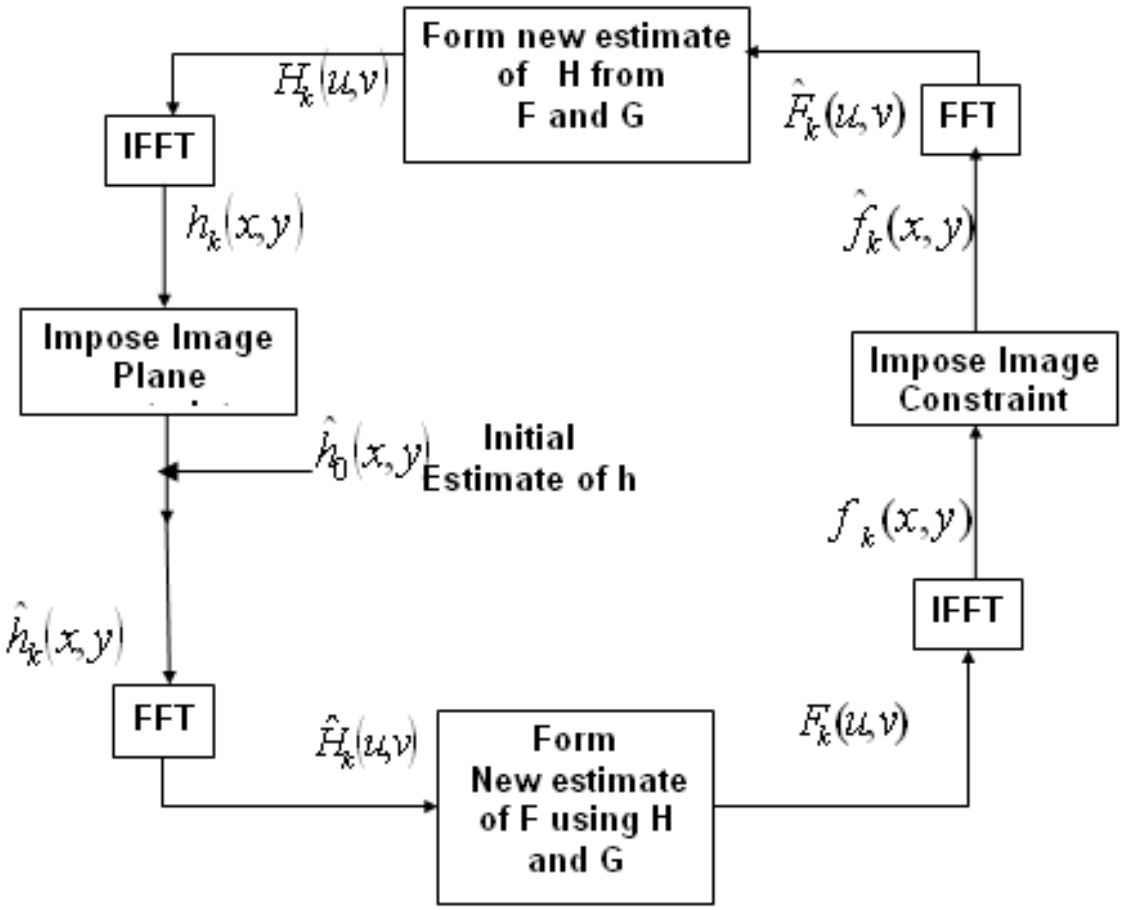

Figure 3.3: The sequence of operations to be performed in the IBD algorithm.

required convolution, $g(x, y)$, have been found.

The image-domain constraint of non negativity is commonly used in iterative algorithms associated with optical processing owing to the non negativity property of intensity distributions. The complete image-domain constraint used in this technique not only forces the function estimate to be positive but also conserves energy at each iteration. The latter condition is realized by uniformly redistributing the sum of the function's negative values over the function estimate.One major problem exists in the above procedure. If $H(u, v)$ contains very small values this will result in very high values in $F(u, v)$ after performing the operation $G(u, v)./H(u, v)$.

The Fourier-domain constraint may be described as constraining the product of the Fourier spectra of f and h to be equal to the convolution spectrum, in agreement with equation(3.9). It should be noted that, at the k^{th} iteration, two estimates for each Fourier

spectrum are available. For example, $H_k(u,v)$ and the estimate $\hat{H}_k(u,v)$ in the as shown in figure(3.3). Both of these estimates have associated properties in common with the desired deconvolved solution. $\hat{H}_k(u,v)$ has a nonnegative inverse transform, and the second estimate $H_k(u,v)$ obviously satisfies the Fourier-domain constraint. Therefore at each iteration the two estimates are averaged to form a composite new estimate as given in equation(3.11). This averaging is not essential for convergence; however, the convergent rate is dependent on β, and a method of selecting the optimum value of β has not been found. Small, confined regions of low or zero value present in $G(u,v)$ are dealt with by using only the estimate $\hat{H}_k(u,v)$. The estimate $H_k(u,v)$ contributes no information to the new estimate. The Fourier-domain constraint can be summarized as follows:
if $|G(u,v)| < Td$

$$H_{k+1}(u,v) = \hat{H}_k(u,v) \tag{3.10}$$

if $|\hat{F}_k(u,v)| \geq |G(u,v)|$

$$H_{k+1}(u,v) = (1-\beta)\hat{H}_k(u,v) + \beta\frac{G(u,v)}{\hat{F}_k(u,v)} \tag{3.11}$$

if $|\hat{F}_k(u,v)| < |G(u,v)|$,

$$\frac{1}{H_{k+1}(u,v)} = \frac{(1-\beta)}{\hat{H}_k(u,v)} + \beta\frac{\hat{F}_k(u,v)}{G(u,v)} \tag{3.12}$$

where $0 \leq \beta \leq 1$. Now the problem of $H_k(u,v)$ (or equivalently of $F_k(u,v)$) having a small modulus is considered. If the modulus of $H_k(u,v)$ is less than the modulus of the convolution spectrum, then instead of performing the linear averaging previously described, the inverses of the two function spectrum estimates are averaged. The new composite estimate is now the inverse of this average. The rationale behind this averaging is simply that large function estimate values, obtained when the inverse filter function has a large value, are prevented from dominating in the average. This is intuitively reasonable, as large inverse filter values are a consequence of small values of $H_k(u,v)$.

3.3.1 Shortcomings of Iterative Blind Deconvolution

The main disadvantage of iterative blind deconvolution is its computational complexity. It takes nearly 300 iterations for getting a useful deconvolved image. Also, there is no convergence criteria defined for IBD. Visual inspection of the deconvolved image has to be used for stopping the iterative process. Another shortcoming of IBD is that, the

convergence of the algorithm is dependent on the initial random estimate of the point spread function.

3.4 IMPROVED ITERATIVE BLIND DECONVOLUTION

An enhanced iterative blind deconvolution scheme is proposed which can solve first two of the above mentioned shortcomings.

3.4.1 Accelerating the convergence

Extremely high values in the Fourier transform of the estimated image as well as in the point spread function was observed as one reason behind the slow convergence of IBD. This problem is overcome by defining *cut-off values* for limiting the maximum value that can appear in the Fourier transform of the estimated image and PSF. This significantly reduces the convergence time.

Also, one more spatial domain constraint is defined for the image. This is the support information explained in section(3.1). One property of point spread functions is that, the sum of all PSF coefficients should be equal to one. This property is used as another constraint in the spatial domain. After each iteration, the sum of all PSF coefficients is made equal to one. This is done by dividing each individual value by the sum of all values. In fact this constraint is observed to be indispensable for getting a satisfactory deconvolved image. The support information results in a faster convergence.

3.4.2 Definition of Stopping criteria

In the original IBD technique the iterative process it stopped by visual inspection of the deconvolved image. This is difficult and may result in wrong judgment. To get rid of this problem, a new stopping criteria based on the image power is defined. Power of the image is defined as the sum of the square of the individual pixel values in the image. In each iteration standard deviation of the image power for last n iterations is calculated. If this standard deviation is less than a particular threshold value TD, then the algorithm can be terminated. The definition of stopping criteria can be summarized as follows:

- Calculate the image power at i^{th} iteration as:

$$pow_i = \sum_{x=1}^{M} \sum_{y=1}^{N} f_i(x,y)^2 \tag{3.13}$$

- Calculate the standard deviation SD of the last $i - n$ power values.

- If $SD < TD$ then stop the algorithm. TD is a predefined threshold value.

3.4.3 The algorithm

The improved iterative blind deconvolution algorithm can be summarized as follows:

1. Get the initial PSF $h_0(x,y)$ with random values. (Size of the PSF must be known before starting the algorithm).

2. - Find $\hat{H}_k(u,v)$ by taking FFT of $\hat{h_k}(x,y)$.
   - Limit the high magnitude values within $\hat{H}_k(u,v)$ as:
     If $\hat{H}_k(u,v) > H_{MAX}$ then $\hat{H}_k(u,v) = H_{MAX}$

3. Compute $F_k(u,v)$ as: $F_k(u,v) = G(u,v)/H(u,v)$

4. Compute the Inverse Fourier transform of $F_k(u,v)$ to get $f_k(x,y)$.

5. Impose the image constraints of non negativity and support on $f_k(x,y)$ to get $\hat{f}_k(x,y)$.

6. - Compute FFT of $\hat{f}_k(x,y)$ to get $\hat{F}_k(x,y)$.
   - Limit the high magnitude values within $\hat{F}_k(u,v)$ as:
     If $\hat{F}_k(u,v) > F_{MAX}$ then $\hat{F}_k(u,v) = F_{MAX}$

7. Compute $H_k(u,v)$ as: $H_k(u,v) = G(u,v)/\hat{F}_k(u,v)$

8. - Compute $IFFT$ of $H_k(u,v)$ to get $h_k(x,y)$
   - Limit the low magnitude values within $h_k(x,y)$ as:
     If $h_k(x,y) < h_{MIN}$ then $h_k(x,y) = h_{MIN}$
   - Make the sum of individual values of $h_k(x,y)$ equal to one, to get $\hat{h_k}(x,y)$ as:
     $SUM_h = \sum_{x=1}^{m} \sum_{y=1}^{n} h_k(x,y)$
     $\hat{h_k}(x,y) = h_k(x,y)/SUM_h$

9.
   - Compute the power of the image as given in equation 3.13.
   - Compute the standard deviation, SD of the power for last t iterations.
   - If $SD < Threshold$ STOP the algorithm. Otherwise goto step 2.

3.5 SIMULATION RESULTS

Simulations were carried out by using MATLAB6.5. Different images and different types of degradations were used in simulation. Some results are given below.

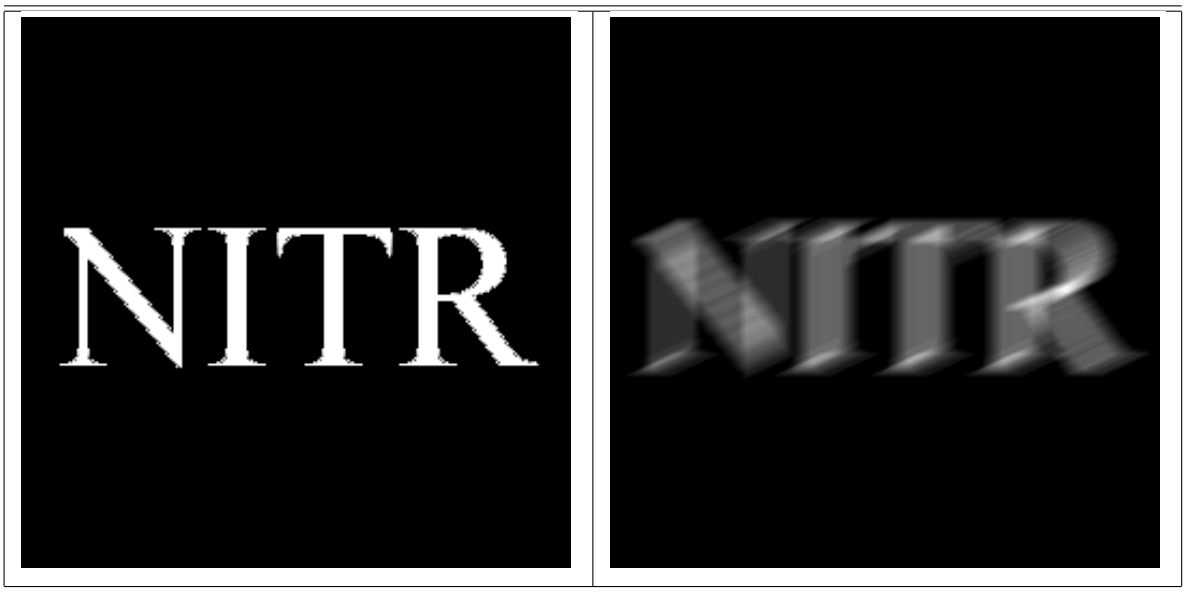

Figure 3.4: (a)Original Image. (b)Motion blured image using 13x23 PSF

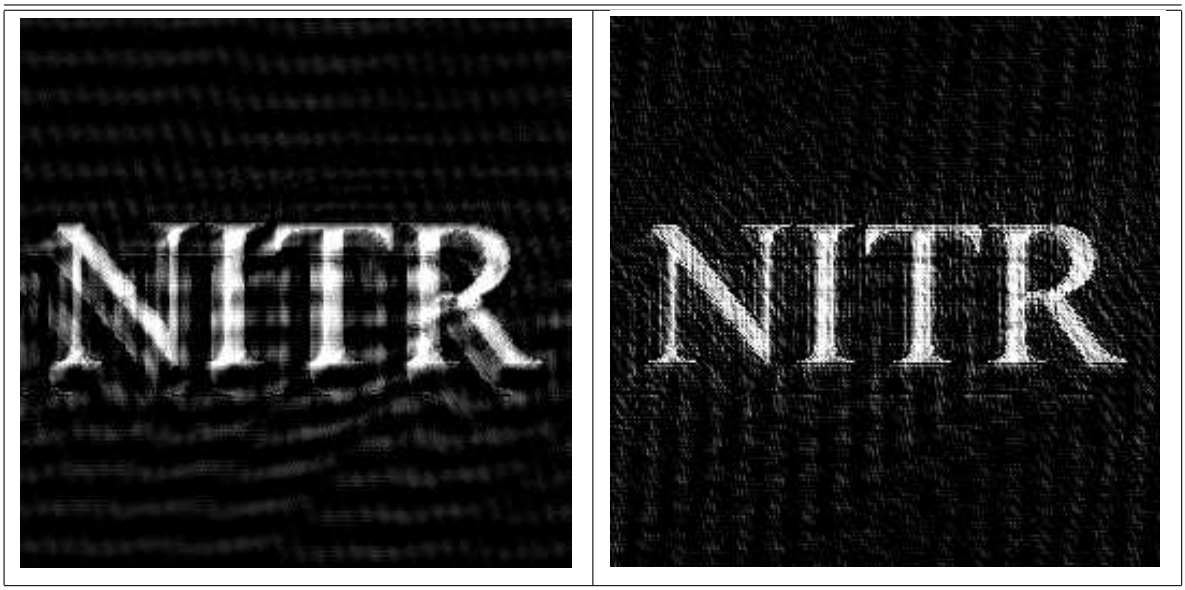

Figure 3.5: (a)Image Deconvolved using IBD with $\beta = .5$. (b)Image deconvolved with IBD with $\beta = 0.1$

Figure 3.6: Results of Improved IBD: (a)Image Deconvolved without support information. $F_{MAX} = 90000$, $DEV = 7.0e + 006$, $H_{MAX} = 1$, h_{MIN}=.0001. Result obtained after 23 iterations. (b)Image deconvolved using IIBD algorithm with support information and same parameters as (a). Result obtained after 21 iterations. The PSNR value obtained is 17.8dB

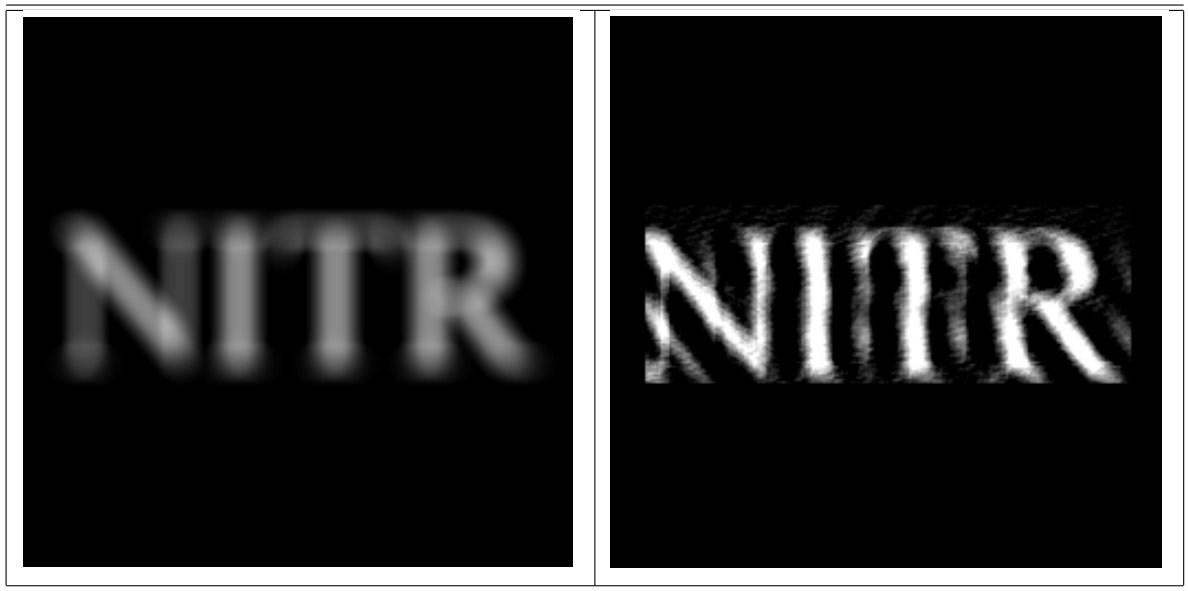

Figure 3.7: Results of Improved IBD: (a)Image blurred with camera defocus PSF of radius 10. (b)Image deconvolved using IIBD algorithm with support information and same parameters as figure(3.5). (b)Result obtained after 168 iterations. The PSNR value obtained is 16.7dB

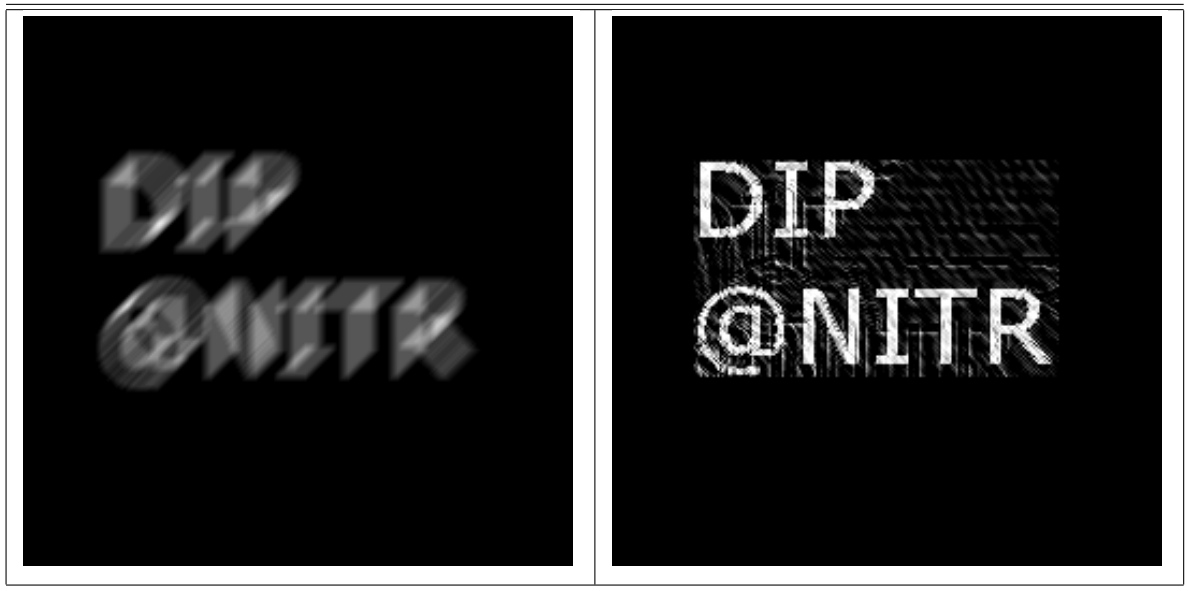

Figure 3.8: Results of Improved IBD: (a)Image blurred with motion blur of length 21 pixels and angle 45 degree. (b)Image deconvolved using IIBD algorithm with support information and same parameters as given in figure(3.5). Result obtained after 29 iterations.

Chapter 4

DENOISING

The main challenge in developing denoising algorithms is to remove noise while preserving image details. Image denoising schemes can be classified broadly into three categories based on the basic methodologies applied to remove noise.

Filtering without Detection

In this type of schemes, the algorithm doesn't try to discriminate noisy pixels from non-noisy pixels. Some operation is defined on the pixels within the filter window. Filter performs the same operation on all pixels within the image while moving from the first pixel to last pixel. Median filter is a very good example of filtering without detection. This kind of filters can successfully filter out noise. But these filters smoothes the edges within the image because of it indiscriminate filtering.

Detection followed by Filtering

This type of filtering involves two steps. In first step it identifies noisy pixels and in second step it filters those pixels. Since filtering is performed only on pixels which are found to be noisy, smoothening of edges will be far less in these techniques.

Hybrid Filtering

In such filtering schemes, two or more filters are suggested to filter a corrupted location. The decision to apply a particular filter is based on the noise level at the test pixel location or performance of the filter on a filtering mask.

4.1 IMPULSE NOISE DETECTION

A switching filter detects noise before performing filtering and filters a particular pixel only if it is noisy. Let x_{ij} and y_{ij} denote pixels with coordinates (i, j) in a noisy and a filtered image, respectively. If the estimated value of the corrupted pixel x_{ij} is $\xi(x_{ij})$, the switching filter concept can be defined by

$$y_{ij} = M_{ij}.\xi(x_{ij}) + (1 - M_{ij}).x_{ij} \tag{4.1}$$

where M_{ij} is the binary noise map, with 1s at the positions of pixels detected as noisy and 0 otherwise. Generally, it is produced by comparing the absolute difference between the original pixel value x_{ij} and some local statistics $\Omega(x_{ij})$ with a threshold T_d as in

$$M_{ij} = \begin{cases} 1 & if \quad |x_{ij} - \Omega(x_{ij})| \geq T_d \\ 0 & if \quad |x_{ij} - \Omega(x_{ij})| < T_d \end{cases} \tag{4.2}$$

The problem of finding the optimal T_d is fairly complex due to its correlation to the image contents, noise probability, and noise distribution. Setting T_d too high, leaves a certain portion of the noisy pixels omitted from the noise map. If is too low, image details will be treated as noise, and the overall image quality will be degraded.

4.2 PERFORMANCE MEASURES

The metrics used for performance comparison of different denoising techniques are defined below

Peak Signal to Noise Ratio (PSNR)

PSNR analysis uses a standard mathematical model to measure an objective difference between two images. It estimates the quality of a reconstructed image with respect to an original image. The basic idea is to compute a single number that represents the quality of the reconstructed image. Reconstructed images with higher PSNR are judged better. Given an original image F of size $M \times N$ pixels and a reconstructed image $\hat{F}$, the PSNR(dB) is defined as:

$$PSNR(dB) = 10 \log_{10} \left(\frac{255^2}{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(Y_{ij} - \hat{Y}_{ij} \right)^2} \right) \tag{4.3}$$

Percentage of Spoiled Pixels (PSP)

PSP is a measure of percentage of non-noisy pixels change their gray scale values in the reconstructed image. In other words it measures the efficiency of noise detectors. Hence, lower the PSP value better is the detection, in turn better is the filter performance. PSP=((Number of non-noisy pixels changed their gray value)/(total number of non-noisy pixels))x100.

Percentage of Hidden Noise(PHN) and Percentage of Faulty Detection(PFD)

Percentage of Hidden Noise is the ratio of Number of undetected noisy pixels to total number of noisy pixels. For an ideal filter PHN should be zero. PHN=((Number of undetected noisy pixels)/(Total number of noisy pixels))$\times$ 100

Percentage of faulty detection is the ratio of the number of pixels wrongly classified as noise to the total number of non-noisy pixels. PFD=((Number of pixels wrongly detected as noise)/(Total number of non-noisy pixels))$\times$ 100.

4.3 IMPULSE NOISE DETECTION BASED ON PIXEL WISE MAD

The main drawback of the median filter is that it modifies pixels not contaminated by noise, thus removing fine details in the image. Therefore, contemporary switching impulse noise filters split the filtering process into two steps: (1) detection of impulses and (2) replacement of impulses with estimated values, where the median is commonly used as the estimator. Detection schemes are based on various concepts: weighted median , rank-order thresholding, local signal statistics , fuzzy reasoning , neural networks ,etc.

The impulse detection concept in pixel-wise MAD, does not require optimizing parameters or previous training. Still, it removes impulse noise very efficiently, while preserving image details. Since both of its steps, detection and estimation, are based on the same median structure, only a simple median filter is required for practical realization. Therefore, the algorithm complexity is equivalent to that of the median filter.The median of the absolute deviations from the median -MAD, is used to estimate the presence of image details, thus providing their efficient separation from noisy image pixels. An iterative

pixel-wise modification of MAD (PWMAD) provides reliable removal of arbitrarily distributed impulse noise.

Most of the known filtering schemes exhibit satisfactory results for salt and pepper noise. However, few of them perform well for random valued impulse noise. PWMAD gives better results for images corrupted with random valued impulse noise.
Let x_{ij}, m_{ij} and d_{ij} represent pixels with coordinates (i, j) of the noisy image, median image and absolute deviation image, respectively. Also let X_{ij}, M_{ij}, D_{ij} denote matrices whose elements are pixels of the corresponding images contained within the $(2K+1) \times (2K+1)$ sized filter window centered at position (i, j). The PWMAD filtering procedure can be described as given below.

1. Find the pixels of the median image as

$$m_{ij} = median(X_{ij}) \tag{4.4}$$

2. Find the pixels of the absolute deviation image as

$$d_{ij} = |x_{ij} - m_{ij}| \tag{4.5}$$

3. Find MAD_{ij} as

$$\begin{aligned} MAD_{ij} &= median(|X_{ij} - median(X_{ij})| \\ &= median(|X_{ij} - m_{ij}|) \end{aligned} \tag{4.6}$$

4. Compute $PWMAD_{ij}$ as

$$\begin{aligned} PWMAD_{ij} &= median(D_{ij}) \\ &= median(|X_{ij} - M_{ij}|) \end{aligned} \tag{4.7}$$

Intuitively we can see that MAD is the median of deviations of pixel values within a filter window. So this can be used as a measure of local variance within the filter window. Higher variance usually is a result of the presence of edges in that area. PWMAD represents a good estimate of the local variance in the presence of impulse noise, i.e., the image details. The absolute deviation image consists of noise and image details eliminated from the noisy image by median filtering. If the details are extracted, only noise remains

and an accurate noise map can be generated. In order to make the robust estimate of image details, the PWMAD image is computed. By subtracting the PWMAD image from image d_{ij}, details are reduced while most of the noise remains. Since the subtraction is performed through the absolute value, positive and negative impulses are treated equally. The whole procedure can be described in an iterative manner as

$$d_{ij}^{k+1} = |d_{ij}^{k} - median(d_{ij}^{n})| \tag{4.8}$$

where d_{ij}^{0} is the initial deviation image. In each step, certain portions of details are eliminated, while noise remains. Due to the fact that most of the details are eliminated after N steps, the noise map is obtained by comparing with a threshold close to zero.

$$NMAP_{ij} = \begin{cases} 1 & if \quad d_{ij}^{N} > Td \\ 0 & if \quad d_{ij}^{N} < Td \end{cases} \tag{4.9}$$

4.4 DOUBLE DERIVATIVE BASED IMPULSE DETECTION

A double derivative based impulse noise removal scheme essentially involves two steps.

1. Detect the presence of an impulse.

2. Replace the pixel by a suitable gray scale value if a pixel is found to be corrupted.

The row wise double derivative can be expressed as:

$$D_R^2 = \begin{bmatrix} D_{1,1}^2 & D_{1,2}^2 & D_{1,3}^2 \\ D_{2,1}^2 & D_{2,2}^2 & D_{2,3}^2 \\ D_{3,1}^2 & D_{3,2}^2 & D_{3,3}^2 \end{bmatrix} \tag{4.10}$$

where $D_{i,j}^2 = Di + 1, j - Di, j$ where $D_{i,j}$ is the single derivative, defined as
$Di, j = xi + 1, j - xi, j$

The following conclusions can be drawn from the above operations.

- $D_{i,j}^2$ is a high magnitude negative quantity if $x_{i+1,j}$ is corrupted by a high value impulse noise.

- $D^2_{i,j}$ is a high magnitude positive quantity if $x_{i+1,j}$ is corrupted by a low value impulse noise.

As detecting the presence of an impulse is more important than its value $abs(D^2_{i,j})$ can be used. But the above method fails if impulses appear continuously in a row, in which case the double derivative does not attain a high magnitude. This shortcoming can be overcome by applying the column wise double derivative after applying the row wise double derivative. A noise map indicating the presence of noisy pixels is obtained after finding the column wise and row wise double derivative. This noise map is used by a median filter to selectively filter only the noisy pixels.

4.5 ITERATIVE NOISE FILTERING WITH EDGE RETRIEVAL

A new simple scheme for impulse noise removal is proposed. The scheme is based on calculating the deviation of values in a straight line passing through the center pixel of the filtering window. The fundamental concept is similar to the double derivative method. But the double derivative method is performed only horizontally and vertically. The diagonal neighbours are not considered in the double derivative.

Another improvement of the algorithm is the *edge detection* which follows the noise detection. In the edge detection phase one unique characteristic of edges is used to retrieve edges which are wrongly detected as noise. The graph given in Fig.4.1 below is completely illustrative of the similarity as well as differences between edges and noise.

It can be seen that edges are also characterized by sudden change of pixel values like noise, but unlike noise the width of the spike in case of an edge is much wider compared to that of a noise. That means, If we take the differences $d1 = x_{i,j} - x_{i,j-1}$ and $d2 = x_{i,j} - x_{i,j+1}$

- For noisy pixel $x_{i,j}$, both $d1$ and $d2$ will be of high magnitude.
- For edge pixel $x_{i,j}$, only one among $d1$ and $d2$ will be high of high magnitude.

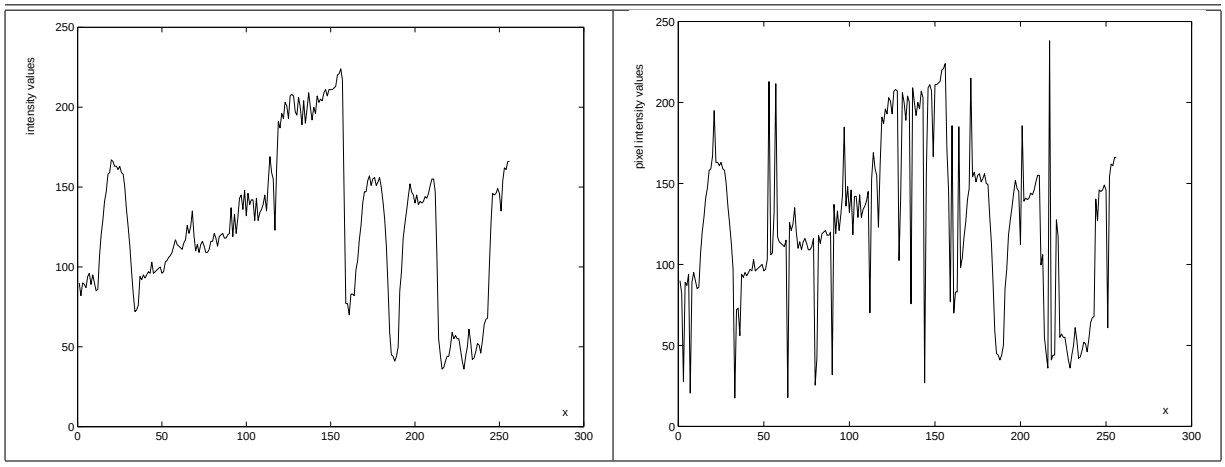

Figure 4.1: (a)Plot of pixel values in a horizontal line of a non-noisy image. (b) The plot for a noisy image.

Also,images taken by a camera contain edges which are slightly blurred due to the imperfections of the imaging device. In that case edges are not abrupt variations instead the variations are gradual. This fact is used to find out pixels which are wrongly classified as noisy. The noise map is modified according to this information.

The proposed algorithm is designed as an iterative scheme in which the image is first filtered with a 3×3 filter window and then filtered with a 5×5 filter window. The threshold value used to decide the presence of noise is initially set to a high value and it is decreased in each iteration. This increases the probability that a noisy pixel is replaced with a value so close to its original value.

4.5.1 Edge retrieval based on single derivative

According to the property of edges depicted in Fig.(4.1) the pixel values gradually increase or decrease while moving through an edge. That is, if the first difference is taken on the edge pixels, all the values will be of the same sign. This fact is used to retrieve edges after the noise detection phase. First derivative is calculated in a 5×5 window, instead of calculating for the whole image. This makes it possible to calculate the first difference through all the lines passing through the middle pixel of a window, instead of just horizontal and vertical.

4.5.2 Edge retrieval based on pixel wise MAD

It has already been observed that $PWMAD$ has very good edge preservation characteristics. So $PWMAD$ can be used as a statistic for retrieving edge pixels after noise detection. It is observed that PWMAD based edge detection outperforms the first derivative based method both in $PSNR$ as well as visual quality.

4.5.3 The Algorithm

The algorithm for Iterative noise filtering and edge retrieval can be summarized as follows:

1.
   - Initialize filter window size $WSZ = 3 \times 3$.
   - Initialize a threshold $TD1$. (70 is a good initial value).
   - If $x_{i,j}$ is the center of the filter window, there are four straight lines passing through that pixel.
     Find the difference between each end pixel and $x_{i,j}$ for each line.
   - Count the number of lines which satisfies the noisy pixel criteria explained in section(4.5).
   - If the the count $CNT >= K1$ then mark the pixel as noisy. (K1=3 gives good results).
   - Make the noise map for all pixels.
   - Compute $PWMAD$ as given in equation(4.7).
   - Compute $E_{i,j} = abs(D_{i,j} - PWMAD_{i,j)}$. If $E_{i,j} > TD3$ mark the pixel $x_{i,j}$ as non-noisy.
   - Decrease the value of $TD1$ as $TD1 = TD1 - C1$
   - Repeat step 1 until $TD1 < t$
2. Perform the operations mentioned in STEP 1, with window size changed to 5x5. Use different threshold values $TD2$, ($TD2 = 30$ gives good result);

4.6 SIMULATION RESULTS

Figure 4.2: (a)Original Lena image of size 256×256. (b) Lena image with 15% random valued impulse noise added to it.

Figure 4.3: (a)Median filtered Image with 15% noise. PSNR=28.35dB(b) Image filtered using double derivative method with Threshold 50. PSNR=28.94dB

Figure 4.4: (a)Lena image filtered using pixel-wise MAD algorithm. PSNR=28.75dB (b) Image (a) after processing by passing the output of PWMAD through the first phase of the proposed algorithm. PSNR=30.16dB

Figure 4.5: (a)Image filtered using the Two-phase scheme with single derivative used for edge retrieval. PSNR=30.18dB (b) Image filtered using the Two-phase scheme with $PWMAD$ used for edge retrieval. PSNR=31.42dB

Chapter 5

CONCLUSIONS AND FUTURE WORK

5.1 CONCLUSIONS

For deconvolution, non parametric techniques give promising results. But use of image constraints alone does not seem to yield a satisfactory restored image. That may be the reason for the unsatisfactory performance of the NAS-RIF algorithm. Also in the NAS-RIF algorithm the minimization of the cost function always does not mean a meaningful solution.

The Iterative Blind deconvolution scheme shows some promise, even though quality of the restored image is not satisfactory. The main disadvantage of IBD is its slow convergence rate. The convergence can be accelerated by limiting the high magnitude values in the frequency domain of estimated image as well as the estimated PSF. It was observed that forcing the sum of the PSF coefficients at the end of each iteration improves the quality of the restored image to a great extend. The original IBD algorithm didn't have a stopping criteria. A new stopping criteria can be defined based on the standard deviation of image power for previous k iterations. The values of the initial PSF does not have any impact on the performance of the algorithm. But the size of the PSF must be known before starting the algorithm.

A priori algorithms for blind deconvolution was also observed to give promising results.

In the cepstrum based motion blur estimation scheme, some excellent results can be obtained for a limited domain of motion blur.

In denoising, pixel-wise MAD was observed to be a very good measure of the edges in the image. So, this statistic can be used for an edge retrieval scheme which follows the noise detection. The edge retrieval scheme improves the PSNR values as well as the detail preservation characteristics of a denoising algorithm.

5.2 LIMITATIONS AND FUTURE WORK

The Improved Iterative deconvolution scheme works well on binary images. But the performance is not satisfactory on gray scale images. This is one area which needs improvement. Also, the algorithm shows variations in performance for different types of point spread functions. These characteristics has to be more thoroughly studied and a more general scheme has to be worked out. A more serious limitation is the requirement that the size of the PSF must be known in advance. Some scheme has to be developed which can at least guess the PSF with some accuracy and later refine it to estimate the size accurately. A priori blur identification techniques may be used for this purpose.

In the denoising based on edge retrieval, the results of the scheme are satisfactory except the time complexity of the scheme. A scheme to adaptively change the threshold values may be designed which can improve the performance.

Bibliography

[1] "A switching median filter with boundary discriminative noise detection for extremely corrupted images," *IEEE Transactions on Image Processing*, 2006.

[2] N. Alajlan and M. Kamel, "Detail preserving impulsive noise removal," *Signal Processing: Image Communication*, 2004.

[3] G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," *Optics Letters*, vol. 13, no. 7, 1988.

[4] J. Biemond, A. M. Tekalp, and R. L. Lagendijk, "Maximum likelihood image and blur identification: a unifying approach," *Optical Engineering*, vol. 29, May 1990.

[5] T. Chen and H. R. Wu, "Adaptive impulse detection using center-weighted median filters," *IEEE Signal Processing Letters*, vol. 8, no. 2, January 2001.

[6] D. G. Childers and D. P. Skinner, "The cepstrum: A guide to processing," *IEEE Proceedings*, vol. 65, no. 10, October 1977.

[7] V. Crnojevic and Z. Trpovski, "Advanced impulse detection based on pixel-wise mad," *IEEE Signal Processing Letters*, vol. 11, no. 7, July 2004.

[8] R. Fergus and B. Singh, "Removing camera shake from a single photograph," MIT CSAIL, Tech. Rep., 2004.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Prentice-Hall India, 2005.

[10] N. Ishihara and S. Komatsu, "Blind recovery of truncated blurred image using adaptive masking method," May 2001.

[11] F. Krahmer, Y. Lin, and D. Widemann, "Blind image deconvolution: Motion blur estimation," August 2006.

[12] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Processing Magazine*, May 1996.

[13] ——, "A novel blind deconvolution scheme for image restoration using recursive filtering," *IEEE Transactions on Signal Processing*, vol. 46, no. 2, 1998.

[14] ——, "Robust classification of blurred imagery," *IEEE Transactions on Image Processing*, vol. 9, no. 2, February 2000.

[15] R. Lokhande and K. V. Arya, "Identification of parameters and restoration of motion blurred images," Indian Institute of Technology, Kanpur, Tech. Rep., 2003.

[16] Z. Lou, "Blind deconvolution using recursive filtering," Master's thesis, University of Illinois at Urbana-Champaign, April 2005.

[17] Z. Mou-yan and R. Unbehauen, "An iterative method of blur identification and image restoration," Institute of Electronics Academia Sinica, Beijing, China, Tech. Rep., 2002.

[18] G. Panda and S. K. Mahapatra, "Efficient filtering of image data corrupted by impulse noise."

[19] W. H. Press and B. P. Flannery, *Numerical Recipes in C*, 2nd ed. Cambridge, 1992.

[20] P. K. Sa, "On the development of impulsive noise removal schemes," Master's thesis, NIT Rourkela, May 2006.