Median Filter Based Digital Image Restoration Using Joint Statistical Modeling

Hankaw Qader Salih

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
August 2018
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Cem Ergün
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Adnan Acan ________________________

2. Asst. Prof. Dr. Cem Ergün ________________________

3. Asst. Prof. Dr. Kamil Yurtkan ________________________

ABSTRACT

Image restoration involves the reduction or complete removal of image degradation in an effort to enhance an image and recover its original form. One of the main methods of image restoration is Joint Statistical Modeling (JSM). This thesis proposes method for image restoration based on JSM and the statistical characterization of the nonlocal self-similarity and local smoothness of natural images. In an effort to improve the image restoration results through JSM, the proposed method involves the addition of a Switching Median Filter (SMF) to JSM and a Median Filter (MF) at the end of every iteration in the restoration process.

Overall, the proposed image restoration method makes the following contributions: it establishes JSM in a domain for hybrid space-transformation; using JSM, it develops a new type of minimization function to be used in solving inverse problems in image processing; and JSM is developing a new rule-based in the Split Bregman method, which is intended to solve any prospective image problems related to a theoretical proof of convergence.

The proposed method was experimentally tested for three kinds of image restoration: image deblurring, image inpainting (text removal), and the removal of mixed Gaussian and salt-and-pepper noise. The results of these experiments indicate that image restoration using the proposed method is a significant improvement compared to conventional JSM. Furthermore, the convergence of the proposed method was also considerably improved relative to JSM.

Keyword: image restoration, joint statistical modeling, image inpainting, image deblurring, noise removal

ÖZ

Resim onarma işlemi, resimdeki mevcut bozunumun azaltılması tamamen ortadan kaldırılması amacıyla yapılan iyileştirme ve asıl haline dönüştürme işlemidir. En başta gelen yöntemlerden bir taneside Ortak İstatiksel Model (OİM) yöntemidir. Bu tezde, resimlerin yerel olmayan özbenzeşlik ve yerel pürüzsüzlük istatiksel nitelendirilmesi ile OİM'e dayalı yeni bir resim onarma yöntemi sunulmuştur. Bu yöntemde OİM'den alınan sonuçları iyileştiremek amacıyla, OİM yöntemine anahtarlamalı ortancı süzgeci eklenerek onarma sürecinde ortancı süzgecin her bir iterasyonda kullanılması öngörülmüştür.

Sonuç olarak, şu katkılar sağlanmıştır; OİM kullanarak resmi tersten işleme problemini çözmede yeni bir azaltma fonksiyonu geliştirilmiştir. Geliştirilen kural bazlı Split Bregman yöntemi ile OİM iyileştirilmiş her türlü olası tersten resim işleme problemlerine kuramsal bir yakınsama kanıtı sunulmuştur.

Önerilen yöntem deneysel olarak üç ayrı resim onarma uygulamasında test edilmiştir; Bunlar, resim netleştirme, resim iç boyama (metin giderme) ve karışık Gauss ve tuz-ve-biber gürültüsünün kaldırılmasıdır. Yapılan deneylerin sonucuna göre resim onarmada önerilen yöntem ile anlamlı bir gelişme sağlanmıştır. Buna ek olarak önerilen yöntem ile yakınsama OİM'den daha iyi olmuştur.

Anahtar kelimeler: resim onarma, ortak istatistiksel modelleme, resim iç boyama, resim netleştirme, gürültü giderme

DEDICATION

To my Son, Mother and Wife

ACKNOWLEDGMENT

I would like to thank God for everything he offered to me. I also especially appreciate my supervisor Asst. Prof. Dr. Cem Ergün, who provided me with the possibility for the success of this thesis. It has been my pleasure working with him.

To my family, especially my mother my wife my brothers my sisters my little son specially my parents.

Finally, I would like to thank all my friends and say a special thank you to every single person who supported me with even a penny for my education. God bless you all.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DSP	Digital Signal Processing
DFT	Discrete Fourier Transformation Technique
DCT	Discrete Cosine Transformation Technique
DWT	Discrete Wavelet Transformation Technique
DWPT	Discrete Wavelet Packet Transformation
FFT	Fast Fourier Transform
FISTA	Fast Iterative Shrinkage-Thresholding Algorithm
FPGA	Field-Programmable Date Array
IDFT	Inverse Discrete Fourier Transformation
IDWT	Inverse Discrete Wavelet Transformation
JSM	Joint Statistical Modelling
LAE	Least Absolute Errors
LAD	Least Absolute Deviation
LSM	Local Statistical Modelling
NLSM	Nonlocal Statistical Modelling
MCA	Morphological Component Analysis
MF	Median Filter
MRF	Markov Random Field
MSE	Mean Square Error
PSNR	Peak Signal to Noise Ratio
PSF	Point Spread Function
SBI	Split Bregman Iteration
SMF	Switching Median Filter

SMW	Sherman-Morrion-Woodbury
TCP	Iterative Framelet-Based Sparsity Approximation Deblurring Algorithm
VLSI	Very Large Scale Integration

Chapter 1

INTRODUCTION

As electronic photographs of a scene, digital images are typically composed of pictorial elements called pixels, which are organized in the formation of a grid. Each pixel contains a particular, quantized value representing the tone at that exact point. Images are captured in a variety of fields, from remote sensing, astronomy, medical imaging, microscopy, down to everyday photography etc. [1].

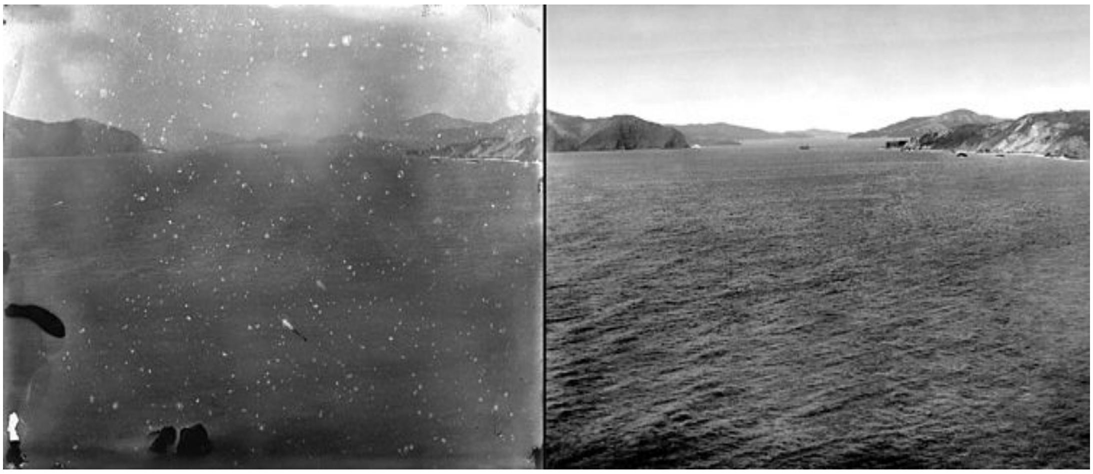
The removal of noise (image restoration) is one of the most critical stages in image processing applications. The process often finds use in many applications – including pattern recognition, image compression, and image encoding – as part of preprocessing. It is possible for an image to become corrupted during any of the preprocessing, acquisition, transmission, and compression processing phases. The corruption of the images usually results from impulse noise caused either by errors with the channel transmission or noisy sensors [2].

Developed in the 1950's, the application of image restoration spans a number of domains including scientific exploration, filmmaking and archrivals, legal investigations, consumer photography, and image and video decoding. It is, however, mostly used in image reconstruction in tomography, radio astronomy, and radar imaging [3].

1.1 Definition of Restoration

The reduction or total removal of degradation in an image is the primary objective of image restoration. As a method of image enhancement, it also involves an attempt at reconstructing the image in its original form. Where both differ is that while image restoration concerns fixing an image blurred for whatever reason, image enhancement works based on human vision and, as such, aims to make the image more appealing.

The methods of image restoration fall into one of two categories: the first includes images where the cause of the degradation is known, while the other is for images for which there is no prior knowledge. For images falling into the former category, a degradation model could be built and subsequently inverted to recover the original image. Figure 1.1 illustrates the result of using image restoration to remove noise from an image [4].



(a) Noisy image (b) Noise removal

Figure 1.1: Image restoration

1.2 Sources of Image Degradation

The degradation process, which is a low pass filter is visually represented as follows:

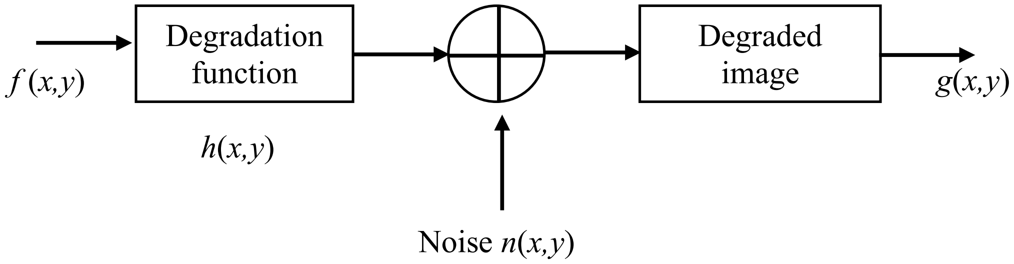


Figure 1.2: Degradation model

Figure 1.2 shows a two-dimensional image f (x, y) and the original input is manipulated on the system h (x, y). Adding noise n (x, y) results in the degraded image g (x, y). The process of digital image restoration is essentially an attempt at approximating the original image f (x, y) from the degraded image [5].

$$g(x,y) = H[f(x,y)] + n(x,y) \tag{1.1}$$

wherein Eq. 1.1, the non-invertible linear degradation operator is represented by the matrix H and the added Gaussian white noise is represented by n. Image denoising and image deblurring are the objectives when H is an identity or a blur operator, respectively. Image inpainting, however, is the aim when H is a mask, a diagonal matrix, the diagonal entries of which are either killing (0) or keeping (1) the related pixels [6].

1.3 Applications of Restoration

Image restoration can be used to rectify a host of potential problems with images in different fields. While the majority of modern applications restrict themselves to dealing with data stored on storage mediums (such as magnetic tape) that have been processed a while after the image's formation, technological advancements in the form of advanced hardware and speedy algorithms have increased the likelihood of real-time restoration.

The restoration of the images returned from the Mariner spacecraft at the California Institute of Technology Jet Propulsion Laboratory in 1960 was one of the first instances of image restoration. The images suffered from geometric distortion, which was attributed to the vision onboard camera [7]. Digital restoration techniques were utilized to remove said distortion and the chosen algorithm found registration reseal marks and calculating a coordinate transformation, which the image was subsequently subjected to overtime. Restoration techniques have found use in the diverse areas of surveillance data (aircraft and satellite imagery), medicine (X rays, acoustic imagery), forensic science (smudged fingerprints), oil exploration (seismic signals), and even in music, as exemplified by Stockham's use of holomorphic deconvolution in the restoration of Enrico Caruso's recordings [8].

1.4 Structure of Thesis

This thesis is structured into 5 chapters. Following this introductory chapter, the second chapter presents a comprehensive review of the literature on Image Restoration using JSM. The third chapter outlines the proposed algorithm, data sets, and an evaluation of the proposed method. The fourth chapter presents and discusses the results of the experiments testing the proposed algorithm. The fifth and final chapter contains the conclusion and recommendations.

Chapter 2

LITERATURE REVIEW

2.1 Image Restoration

Image restoration is an essential key concern in image processing. The main aim is to recover an image from a distorted version such as blurred, painted and noise image. These basic restoration techniques will be introduced in the following sections.

2.1.1 Image Deblurring

The process of deblurring involves the removal of the blur, such as motion blur or defocus aberration, from images. The blur is usually modeled after the convolution of an (occasionally space/time-varying) Point Spread Function (PSF) related to a supposedly sharp input image, where the PSF and the intended sharp input image for recovery are unknown [9]. The degradation of the image is computed as:

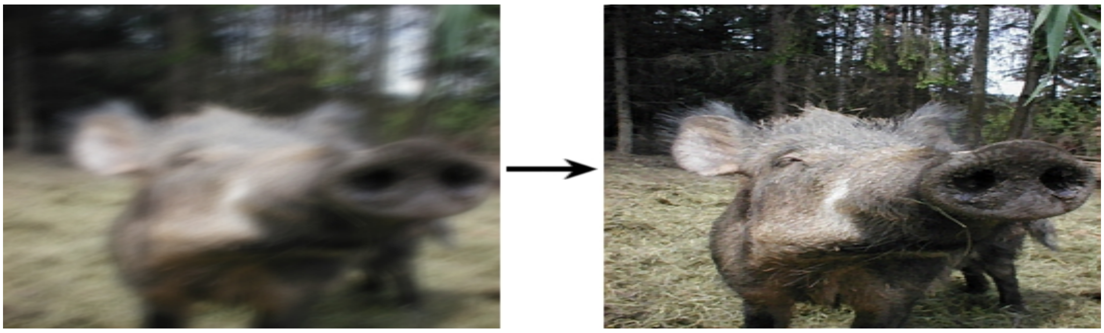
$$y = h * x + w \tag{2.1}$$

where in Eq. 2.1, the original image and the degraded image are represented by x and y respectively, the additive noise is represented by w (white Gaussian noise is taken); the blurring operator's PSF by h, and * represents the mathematical operation of convolution and could alternatively be denoted by its spectral equivalence. Similarly, the following equation is the result of the application of DFT to Eq. 2.2:

$$Y=H*X+W \tag{2.2}$$

where in Eq. 2.2, the Fourier Transforms of *y*, *x*, *h* and *w* are represented by their capital letters. However, due to the fact that the DFTs restoration filters are typically the result of spectral representation when properly executed, it is often the preferred choice. Conversely, noise explosion results from using the PSF of the blurring filter to divide the Fourier Transform [10].

The problem of deblurring an image is inherently challenging as the blurred image we observe only partly contains the solution, thus resulting in added constraints as there is an infinite number of images and blur kernels that can be combined to create in the observed blurred image. There are also numerous sharp images that, even when the blur kernel is known, could possibly match the observed blurred and noisy image following their convolution with the blur kernel. Figure 2.1 shows an image (a) with an added blur and its deblurred image (b) [10].



(a) Blurred image

(b) Deblurred image

Figure 2.1: Image deblurring

2.1.1.1 Blurring

Image blurring is caused by a number of different reasons, including optical aberration, atmospheric scatter, lens defocus, and temporal and spatial sensor integration. While it is also easily noticed by the human eye, the creation of a metric for estimating the blur in images has proved to be a considerable challenge since relatively little is known about the relevant processing mechanism [11].

Uniform Blur

Uniform blur is one method of removing specks and noise from an image and is used when the noise covers the image entirely [11]. The blurring of this kind usually moves either vertically or horizontally and can be circular with a radius *R*, calculated as:

$$R= \sqrt{g} +f \tag{2.3}$$

where in Eq. 2.3, g is the horizontal size blurring direction, *f* is the vertical blurring size direction, and *R* is the radius size of the circular average blurring. As in Figure 2.2, shows the matrix of the uniform blur.

Box blur (Uniform blur) $\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\1 & 1 & 1\\1 & 1 & 1\end{bmatrix}$

Figure 2.2: Matrix of uniform blur

Gaussian Blur

The Gaussian blur effect is a filter, which is used to incrementally blend a specific number of pixels in the pattern of a bell shape. The resulting blurring is concentrated in the center; thus, it is less prevalent around the edges as shown in Figure 2.3. A Gaussian blur is applied to an image when the objective is maximum control over the level of blurring [12,13].

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

(a) Gaussian blur (3x3) (b) Gaussian blur (5x5)

Figure 2.3: 3x3 and 5x5 matrix representation of Gaussian blur

Motion Blur

Another filter is the motion blur effect, which involves adding blur in a specific direction, such that it makes the image appear as though it is moving. Depending on the specific computer program used, the motion blur can either be controlled by pixel intensity, distance (0 to 255), or by direction or angle (–90 to +90 or 0 to 360 degrees) [11,12]. As in Figure 2.4, shows the matrix of motion blur.

Motion blur (3*3) $\frac{1}{9}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Figure 2.4: Matrix of motion blur

2.1.2 Image Inpainting

The process of reconstructing parts of images and videos that have been lost or damaged is known as inpainting. In the world of museums, such a task would be left to the expertise of an experienced art restorer or conservator. In the digital world, however, the task of inpainting (image or video interpolation) is carried out using complex algorithms that substitute the image data’s corrupted parts (primarily minute defects or regions). As a technique for transforming a seemingly undetectable image form, inpainting is as old as art itself. It is used for a variety of reasons and in a similarly wide range of applications, including, but not limited to, the replacement/removal of selected objects and the restoration of damaged paintings and photographs. Inpaining is geared towards the total reconstitution of the artifact in question in an effort to restore its unity and improve its legibility [14].

How the gap is to be filled is determined using the global picture with the primary objective of restoring the image's unity. The structure of neighboring gaps is intended to fade into the affected gap at the same time as contour lines at the boundaries are extended into the gap. The various regions in the gap's interior, represented by the contour lines, are filled with colors corresponding to the boundaries and the smaller details are painted on to add texture.

Structural Inpainting

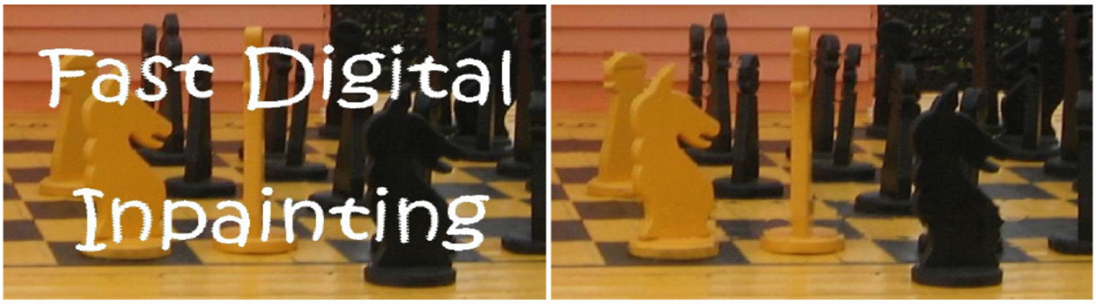
Structural inpainting utilizes a geometric approach to fill in the information missing from the intended inpainting region. The algorithms center on the consistency of the geometric structure.

Textural Inpainting

As with everything, the methods of structural inpainting have their advantages as well as disadvantages. The primary concern, however, is that they are not all capable of restoring texture. The reason for this is that a missing portion cannot be restored by simply extending the surrounding lines into the gap since texture has a repetitive pattern.

Combined Structural and Textural Inpainting

A combination of textural and structural inpainting techniques attempt to simultaneously texturally and structurally fill in the missing image data as many of an image's parts have both structure and texture. The boundaries separating the regions of an image accumulate a host of structural information in a complex fashion that results from the blending together of distinct textures. It is for this reason that innovative inpainting techniques represent an attempt at combining textural and structural inpainting, Figure 2.5 below illustrates the removal of an image inpainting [15].



(a) Painted image (b) Restored image

Figure 2.5: Image textural inpainting

2.1.3 Image Noise

Image noise refers to random variations in the color information or brightness in images; it is typically a feature of electronic noise. While image noise is usually created by the sensor and circuitry of a digital camera or scanner, it can similarly occur in film grains and the inevitable shot noise of a model photon detector. Image noise remains an unintended consequence of capturing images and involves the addition of bogus, inessential information to the image(s) in question [16].

2.1.3.1 Impulsive Noise

Impulsive noise is made up of a series of short pulsating "on/off" noise; it is caused by a number of sources, including a communication system's opposing channel environments, switching noise, surface degradation or dropouts in audio recordings, computer keyboard clicks, amongst others [16].

Consequently, an impulsive noise filter is used to enhance the intelligibility and quality of noisy signals, and to improve the strength of adaptive control systems and pattern recognition. Median filters are the conventional impulsive noise removal method; however, they usually cause a degradation of the original signal [17].

Salt and Pepper Noise

“Salt” and “pepper” noise results when corrupted pixels adopt either the minimum value of 0 or the maximum value of 255 as this results in black and white spots in the image. It is necessary to note that such noise, in whatever form it manifests, is first removed from the image before it undergoes any further processing [18]. “Salt” and “pepper” noise, also known as intensity spikes, is a type of impulsive noise. It is typically the result of analog-to-digital converter errors, dead pixels, errors in data transmission, faulty memory locations, pixel element malfunctions in the sensors of the camera, or digitization process timing errors. As can clearly be seen from Figure 2.6, the probabilities are taken between the minimum (P_a) and maximum (P_b).

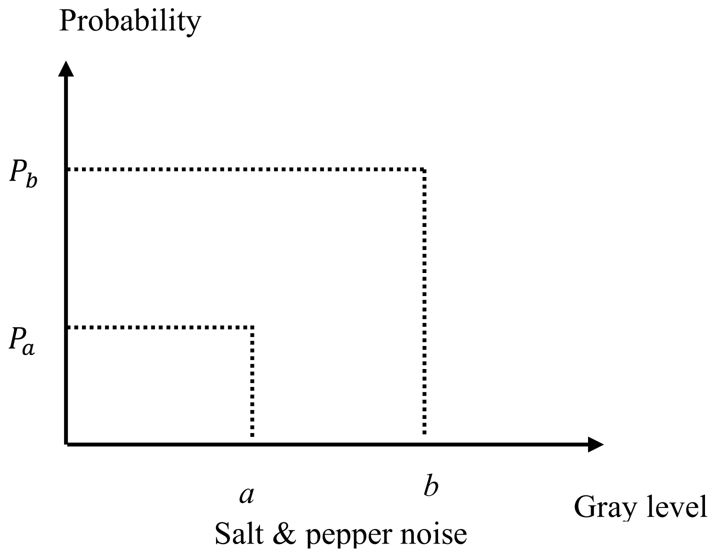
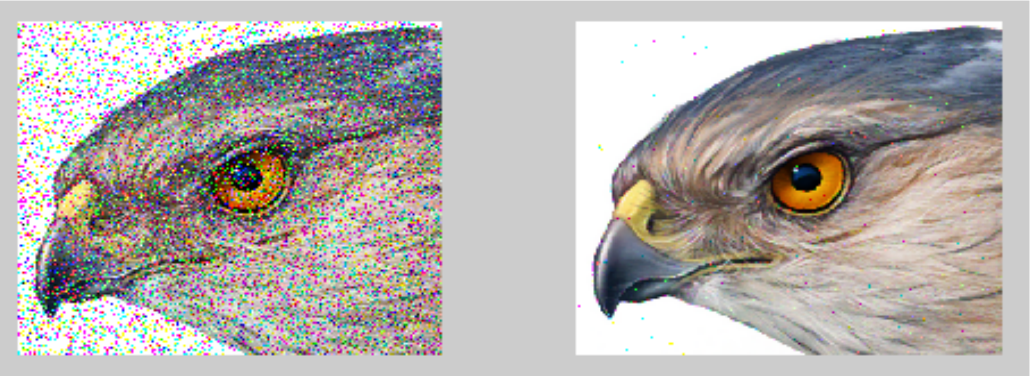


Figure 2.6: PDF for salt and pepper noise model

$$P(g) = \begin{cases} P_a & for \quad g = a \text{ (Pepper)} \\ P_b & for \quad g = b \text{ (Salt)} \\ 0 & \text{otherwise} \end{cases} \tag{2.4}$$

where in Eq. 2.4, $P(g)$ is distribution of salt and pepper noise in the image, a and b denotes level of gray. So if $b > a$, gray level b will appear as a light dot in the image and level a will appear like a dark dot. If either P_a and P_b are zero, the impulse noise is called unipolar noise. Then if neither P_a and P_b are zero and if they are approximately equal the impulse noise called “salt and pepper”. The typical intensity values are 0 for “pepper” noise and 255 for “salt” noise in an 8-bit image. Figure 2.7 below provides an illustration of “salt” and “pepper” noise [19].



(a) Noisy Image (b) Restored Image

Figure 2.7: Removal of salt and pepper noise

2.2 Image Enhancement

The process of enhancing an image includes some variety of techniques aimed at either improving an image’s visual form or converting it to an entirely different from where it can easily be analyzed by humans and machines alike. The image enhancement process involves improving the quality of the image even without knowing how it came to be degraded. It is a procedure used to improve how easily the image can be perceived or interpreted by its human viewers [20]. It is also aimed at making the quality of the resulting image superior to the original, especially in relation to its intended purpose as the quality of a resulting image might be reduced, relative to the original, when it is converted from one form to another through transmission, scanning, imaging, and other similar processes [21].

2.2.1 Spatial Domain Techniques

The pixels of an image are handled individually using spatial domain techniques. These techniques – such as histogram equalization, logarithmic transforms, and power law transforms – enhance images by directly and systematically altering the values of their pixels. They are also suitable for improving the overall contrast of the image as they allow the gray values of single pixels to be altered accordingly. Despite their individual effect on the image's pixels, they improve the overall quality of the image, which can sometimes lead to unintended results [22].

Log Transformation Technique

One of the simpler spatial domain image enhancement techniques is log transformation. It is used primarily to improve the contrast of darker images. Essentially a grey level transform, the technique involves altering the grey levels in the pixels of the image. The transformation process used involves generating a broader range of low grey output level values from an initial, narrower, range of values [20]. The log transformation technique, in its general form, is presented mathematically using the following formula:

$$S=c\ log(1+r) \tag{2.5}$$

where in Eq. 2.5, c is an arbitrary positive constant, r and s are the intensities of the original and transformed images, respectively, with intensity profile 0 through 255.

Power Law Transformation Technique

Another widely used grey level transformation technique, power law transformation shares a number of conceptual similarities with frequency-domain-based alpha rooting to the extent that it involves increasing the input grey level by a particular numeric power [23]. It is also operationally similar to log transforms since power law transforms having γ fractional values are mapped into a limited variety of output levels, thus resulting in increased contrast. It is mathematically represented as follows:

$$S = br^{\gamma} \tag{2.6}$$

where in Eq. 2.6, b and γ are the arbitrary positive constants, r and S are the intensities of the original and transformed images.

2.2.2 Transform Domain Techniques

Frequency domain or transformation techniques – rather than affect the image directly – alter the orthogonal transform of the image instead. As such, they are more fitting for image processing with a focus on the frequency content of the image in question. As a method of image enhancement, frequency domain techniques are based on computing a discrete, unitary, 2-D transform of the image; for example, implementing an inverse transform after the transform coefficients have been altered using the operator M in the case of the 2-D DFT. Two components: the magnitude and phase, are used for the orthogonal transform. While phase allows for the return of the image to the spatial domain, magnitude is comprised of the image's frequency content. A conventional transform domain allows for the manipulation of the image's frequency content, thus permitting the enhancement of high-frequency content (e.g. edges) and other subtle elements [23].

2.2.3 Discrete Fourier Transform (DFT)

The Discrete Fourier Transform (DFT) is used in mathematics to convert a fixed series of equidistant samples in a function to complex-valued frequency function in the form of a series of equidistant samples of the discrete-time Fourier transform (DTFT) having the same length. The DTFT is sampled at an interval equal to the length of the input sequence. A reverse DFT is a Fourier series that utilizes DTFT samples as the coefficients of complex sinusoids and the DTFT frequencies to which they correspond. It also uses sample-values identical to those found in the input sequence, making it a representation of the latter in the frequency domain. The DTFT of the input sequence is continuous (as well as periodic) if the sequence spans every possible non-zero value of a function, while the DFT provides a discrete sample of one cycle. Conversely, the DFT of a once-cycle sequence produces the none-zero values of the DTFT cycle [27].

As the foremost type of discreet transform, the DFT is used to conduct Fourier analysis in a variety of contexts. Function samples can take the form of pixel values in a row or column of a raster image as in image processing. They can also take the form of a time-variable signal or quantity like a radio signal, sound wave pressure, or daily temperature readings over regular intervals, as they do in digital signal processing. The DFT can further be used to solve partial differential equations and a variety of other operations like the multiplication of large integers and convolutions efficiently [25].

The DFT is easily executed on computers through either dedicated hardware or numerical algorithms due to its use of a limited amount of data. Such computer-based implementations typically utilize fast Fourier transform (FFT) algorithms. In fact, this is so often the case that the terms “DFT” and “FFT” are used synonymously. Previously, however, “FFT” was used to refer to the more ambiguous expression (finite Fourier transform) [28].

Fast Fourier Transform (FFT)

Both the DFT and IDFT are computed using a FFT algorithm. The algorithmic analysis transforms a signal received in its original form (typically space or time) into its corresponding frequency representation and vice-versa. It does so and rapidly compute the transformations by factoring the DFT matrix into a collection of sparse (primarily 0) factors [28].

Of particular importance where frequency (spectrum) analysis is concerned, the DFT produces a discrete frequency representation from a discrete time-based signal. Calculating the Fourier transform using either a DSP-based system or a microprocessor would be nearly impossible in the absence of a transform that produces a discrete-frequency signal from a discrete-time signal.

The computing process in initial DFT methods was exceedingly time-consuming. Consequently, FFT emerged as a way to reduce the computing time, making it possible to surmise that FFT is simply the algorithmic computation of DFT in a manner such that it shortens the computational stage(s).

2.3 Convolution Method

Convolution can be described as the method of blurring out colors into the missing areas in an image. Mathematically identical, both repeated blurring and diffusion are founded on the notion of isotropic diffusion. The colors of each pixel in a blurry image are averaged with a small section of the colors in surrounding pixels.

The primary aim of convolution is to form some kind of kernel (an *N* by *N* matrix) that is repetitively convolved over the image and fills in the necessary pixels on the basis of the values of their surrounding pixels. The manner in which the colors are spread into the corrupted space(s) from the surrounding areas is determined by values in the matrix. The process of spreading the colors is repeated as many times as necessary until the entire image has been restored to its original form. Additionally, this method is particularly advantageous as it involves only repeated multiplication [29].

The following example shows how to compute the 2-D discrete convolution of two input matrices. Figure 2.8 shows the first input matrix (I1) representing an image and is denoted as:

$$\mathrm{I1} = \left[\begin{matrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{matrix}\right]$$

Figure 2.8: First matrix (I1)

Figure 2.9 shows the second input matrix (I2) representing another image and is denoted as:

$$I2 = \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

Figure 2.9: Second matrix (I2)

The matrixes given above in Figure 2.8 and Figure 2.9 shows the process by which the (1,1) output element (zero-based indexing) can be produced using these steps:

1. Rotate the second input matrix, I2, 180 degrees around the element at its center.
2. Reposition the element at the center of I2 such that it lies on top of the (0,0) element in I1.
3. Multiply each element of the rotated I2 matrix using the I1 element below.
4. Add up the individual products gotten in the above third step.

Thus, the (1,1) output element is calculated as:

0*2+0*9+0*4+0*7+17*5+24*3+3+0*6+23*1+5*8= **220**, this calculating (1,1) output of Convolution show in the Figure 2.10 [30].

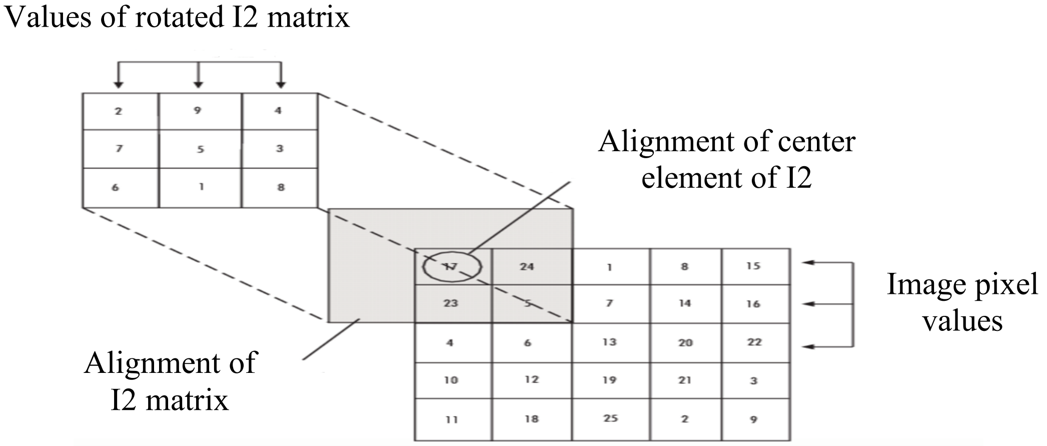


Figure 2.10: Convolution method

2.3.1 Blind Deconvolution Algorithm Technique

The blind deconvolution algorithm is particularly useful when no other information is known about the distortion (noise and blurring). It is used to simultaneously refurbish the image and the PSF. Each iteration uses the speed-up, restrained Richardson Lucy algorithm. The characteristics of optical systems, such as cameras, double as supplementary input parameters with the advantage of improving the quality of the resulting post-restoration image. The constraints of PSF could be conveyed using a function specified by the user. The blind deblurring method is mathematically denoted as:

$$g(x, y) = PSF * f(x,y) + \eta(x,y) \tag{2.7}$$

where in Eq. 2.7, the observed image is represented by $g\ (x, y)$, the constructed image by $f\ (x,y)$, and the additive noise term by $\eta\ (x,y)$ [29,31].

Two kinds of devolution methods exist, maximum likelihood restoration and projection-based blind devolution. The latter involves the simultaneous restoration of the true image, as well as the PSF. The resulting process begins, first, by making projections of the PSF and the true image, in that order, and is cylindrical. The process is repeated severally until a specific previously-determined convergence condition has been met. In addition to its insensitivity to noise, one other benefit of this particular process is that it seems to be consistent even in the face of support-size inaccuracies. Conversely, it is problematic in that it is hardly unique and has been known to result in local minima-associated errors [29].

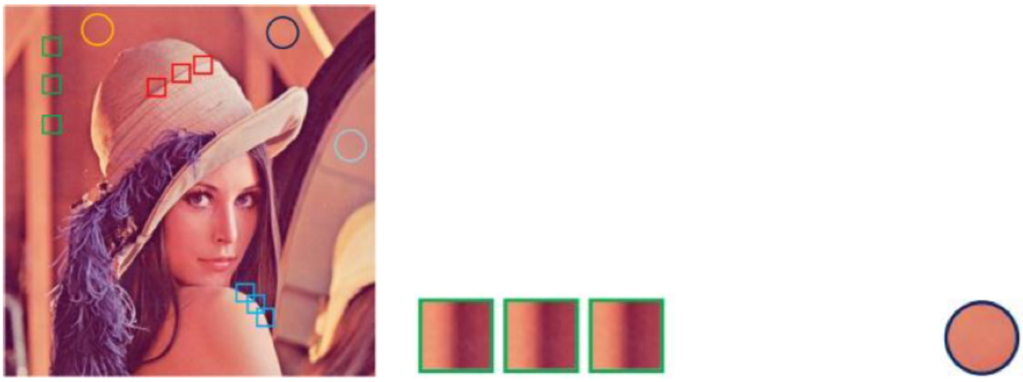
The second approach (maximum likelihood restoration) involves the approximation of the maximum likelihood of the covariance matrices and PSF. Other factors, such as symmetry and size also need to be taken into account since the resulting PSF estimate is not unique. The approach is advantageous in that it involves a minimal level of computational complexity and also allows for the discovery of the noise, blur, and power spectra of the true image [32].

2.4 Joint Statistical Modeling (JSM)

A holistic approach to nonlocal self-similarity and local smoothness makes it possible to define a unique JSM via the combination of Nonlocal Statistical Modeling (NLSM) at the block level in the transform domain and Local Statistical Modeling (LSM) for smoothness in the space domain at the pixel level. This combination is mathematically expressed as Eq. 2.8, [6].

$$\psi_{LSM}(\mathcal{U}) = \tau .\, \psi_{LSM}(\mathcal{U}) + \lambda \,.\, \psi_{NLSM}(\mathcal{U}) \tag{2.8}$$

As such, the trade-off between two competing statistical terms in Eq. 2.8 is controlled by the regularization parameters τ and λ. NLSM corresponds to the nonlocal self-similarity above and maintains image nonlocal consistency, effectually retaining the sharpness and edges. LSM on the other hand corresponds to the local smoothness above and maintains image local consistency, effectively suppressing noise. Figure 2.11 provides an illustration of the image restoration process as carried out using the JSM [6].



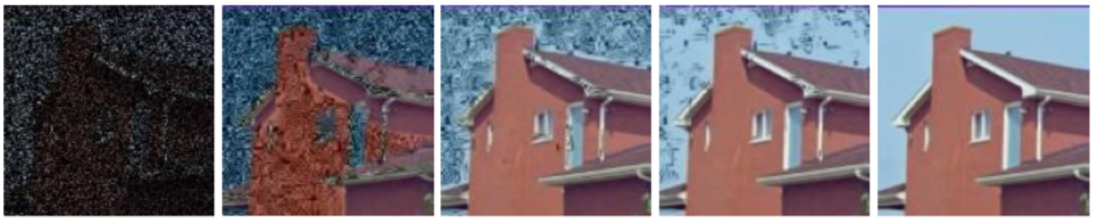
(a) (b) (c)

Figure 2.11: Illustrations for (a) Natural images (c) Local smoothness, (b) Nonlocal self-similarity.

$$\psi_{JSM}(\mathcal{U}) = \tau .\, \psi_{LSM}(\mathcal{U}) + \lambda \,.\, \psi_{NLSM}(\mathcal{U}) = \tau .\, \|\mathcal{D}\mathcal{U}\|_1 + \lambda \,.\, \|\Theta_{\mathcal{U}}\|_1 \tag{2.9}$$

In Eq. 2.9 τ and λ are regularization parameters, above, JSM denotes both the nonlocal self-similarity ($\Theta_{\mathcal{U}}$) and local smoothness ($\mathcal{DU}$) of natural images, while also combining the benefits of both. Consequently, the Split Bergman Iterative (SPI)s developed to make JSM tractable and robust, and to resolve any problems that may arise when effectively optimizing JSM as a regularization term. The implementation of the JSM regularization term and proof of convergence are detailed in the next chapter. Furthermore, the results of extensive experiments also attest to the validity of JSM [6].

Figure 2.12 (a) is the degraded image of House with 20% of the original sample, i.e., Ratio=20%. As the iteration number k increases, it is evident that the quality of the restoration image also increases as can be seen in Figure 2.12 (b)-(e) [6].

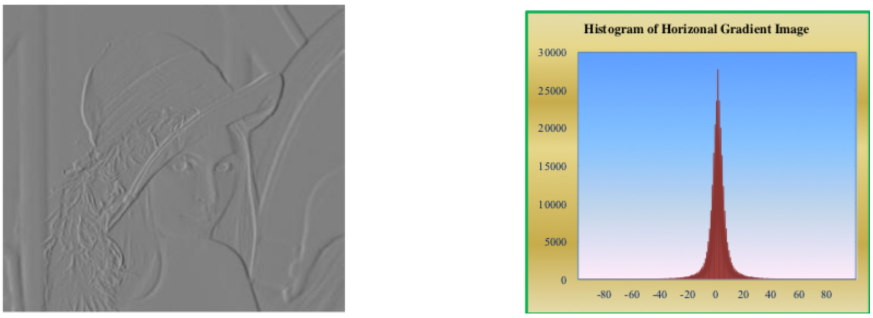


(a) *k*=0 (b) *k*=60 (c) *k*=120 (d) *k*=210 (e) *k*=300

Figure 2.12: Image restoration process for different values of *k*

2.4.1 Local Statistical Modeling (LSM)

The closeness of neighboring pixels in an image's two-dimensional space domain is described as local smoothness, which also implies that neighboring pixels share similar intensities. Although many models can be used to aid the characterizing image smoothness, a mathematical formulation of LSM for 2D space domain smoothness is used for present purposes. Figure 2.13 shows illustrations of local statistical modeling for smoothness in a space domain at the pixel level: (a) Gradient picture in the horizontal direction of image Lena; (b) Distribution of horizontal gradient picture of Lena [33].



(a) (b)

Figure 2.13: Illustrations for LSM

The horizontal and vertical finite difference operators respectively represented as $\mathcal{D}_h = [1 \quad -1]$ and $\mathcal{D}_V = [1 \quad -1]^T$, are the most commonly used filters. The gradient picture in the horizontal direction of the image Lena and its corresponding histogram are shown in Figure 2.13 above. Figure 2.13 also reveals a very narrow distribution of pixel values, which is mostly close to zero. The statistics of both filters mentioned above are modeled using a Generalized Gaussian Distribution (GGD) [34], which is denoted as:

$$\mathcal{P}_{GGD}(X) = \frac{V.\eta(v)}{2.\Gamma(1/_V)} . \frac{1}{\sigma_X} e^{-[\,\eta(v).|X|/\sigma_X]^V} \qquad (2.10)$$

where $\eta(v) = \sqrt{\Gamma(3/_v)\Gamma(1/_v)}$ and $\Gamma(t) = \int_0^\infty e^{-u}u^{t-1}\, du$ are a gamma function, the standard deviation is represented by σ_X and *v* is the shape parameter. If *v=2,* then the distribution $\mathcal{P}_{GGD}(X)$ is a Gaussian distribution function, if *v =1*, it is a Laplacian distribution function, and lastly, it is a hyper-Laplacian distribution if *0<v <1* (see [35] for a more intricate discussion on the values of *v*).

The Laplacian distribution was chosen to model the marginal distribution of gradients in natural images as part of a trade-off between accurate image statistics modeling and the efficient solution of the optimization problem. As such, we take D = $[\mathcal{D}_v; \mathcal{D}_h]$ and v = 1 in Eq. 2.11 to calculate LSM at the pixel level in the space domain. The conforming regularization term ψ_{LSM} is expressed as:

$$\psi_{LSM}(\mathcal{U}) = \|\mathcal{D}\mathcal{U}\|_1 = \|\mathcal{D}_v\mathcal{U}\|_1 + \|\mathcal{D}_h\mathcal{U}\|_1 \quad (2.11)$$

2.4.2 Nonlocal Statistical Modeling (NLSM)

Local smoothness is an important consideration when it comes to natural images. Also important is nonlocal self-similarity, which represents the textural and structural uniformity of natural images in a nonlocal area and can be used for effectively maintaining sharpness and edges to keep the nonlocal image consistent. The traditional nonlocal regularization terms covered in the first section, however, use a weighted approach when characterizing self-similarity through the introduction of a nonlocal graph that trails the level of similarity between the blocks. This approach is infamous for its inability to recover more accurate structures and finer image textures.

Recent attempts at transforming a 3D array of similar patches and decreasing their coefficients have led to commendable results in the areas of the image and video denoising [36,37].

The 3D array generated by stacking identical image patches needs to be transformed to determine the transform coefficients' distribution. This distribution, in turn, is used to mathematically represent natural images' nonlocal self-similarity. As such, this model is termed NLSM for 3D transform-based self-similarity.

In a 3D transform domain, the NLSM for self-similarity is mathematically denoted as:

$$\psi_{NLSM}(\mathcal{U}) = \|\ominus_u\|_1 = \sum_{i=1}^{n} \left\|T^{3D}(Z_{u^i})\right\|_1 \tag{2.12}$$

The convexity of NLSM in Eq. 2.12 can be technically justified as follows: to make it clear, define R_i^{3D}as the matrix operator that extracts the 3D array Z_{u^i} from $u,$ i.e., $Z_{u^i} = R_i^{3D}\ u$. Then, define $G_i^{3D} = T^{3D}R_i^{3D}$, which is a linear operator. It is important to observe that $\left\|T^{3D}(Z_{u^i})\right\|_1 = \left\|\ G_i^{3D}\ u\right\|_1$is convex with respect to u. Since the sum of convex functions is also convex, Eq. 2.12 is convex as to u.

The main benefit offered by NLSM is that it exploits the self-similarities of image blocks that are globally positioned in a more statistically efficient manner in the 3D transform domain, as opposed to nonlocal regularization-incorporated graphs [38].

2.12 Related Works

The Nonparametric Bayesian methods proposed by Mingyuan et. al. [39] are used to recover images based on their specific incomplete, noisy, and/or compressive measurements. By way of an abridged Beta-Bernoulli process, they suggest an appropriate dictionary for the image under recovery and the corresponding data being tested. The use of these learned dictionaries has resulted in significant advancements in image recovery where compressive sensing is concerned, relative to the use of standard orthonormal image expansions. Furthermore, just as projections for the compressive measurement are specifically adjusted for the learned dictionary, so also are basic (incomplete) measurements determined using a subset of randomly-selected uniform image pixels. Dirichlet and probit stick-breaking processes are used to exploit the existing spatial interrelationships within imagery, for which numerous examples and comparisons can be found in the literature.

The method proposed by Elad et. al. [40] presents a new algorithm for inpainting that fills in overlapping texture holes and the image layers of cartoons. The algorithm was directly adapted from Morphological Component Analysis (MCA), which is a new method of sparse, representation-based image decomposition intended for the separation of cartoon layers and combined linear texture in an image [40], redundant multiscale transforms, and their application for morphological component analysis. The method involves the natural fitting of missing pixels into the framework of separation, thus resulting in separate layers (as a by-product of the inpainting process). The method considers hole-filling, separation, and denoising to be a single, unified task in contrast to that proposed by Bertalmio et al [40], which takes the decomposition and filling stages as two separate tasks in a wider system.

Hiroyuki et. al.'s [41] method is a generalization of the tools and results used in the fields of image reconstruction and processing. The generalization is done in reference to the non-parametric statistics field. More specifically, the ideas underlying kernel regression is adapted and expanded for use in upscaling, fusion, interpolation, and denoising, amongst others. In an effort to illustrate how several existing algorithms like the popular bilateral filter are simply special cases of their proposed framework, the authors drew parallels between other existing methods and theirs while also providing practical illustrations of the resulting algorithms and analyses.

The method outlined by Yan Ran et.al. [42] was intended to rectify the problem of restoring images that had been subject to impulse and Gaussian noise. While conventional solutions for these kinds of noise are geared towards the minimization of an objective functional with a ℓ^1 fidelity term and a MumfordShah regularizer, the proposed algorithm aimed to minimize an entirely new objective functional, which has a content-dependent fidelity term that incorporates ℓ^1 and ℓ^2 norm-measured fidelity terms. The functional's regularizer is a product of the ℓ^1 norm of the underlying image's constricted framelet coefficients. The filters of these coefficients are used in extracting images' geometric features. An Iterative Framelet-Based Sparsity Approximation Deblurring Algorithm (IFASDA), who are automatically determined parameters adaptively vary at each iteration, is then proposed for the functional. As such, IFASDA can be considered to be a parameter-free algorithm, making it a more realistic and appealing choice. Its efficacy can be seen in how it handles the deblurring of images corrupted by impulse and Gaussian noise, in addition to enhancements in visual quality and PSNR relative to other existing methods. Additionally, an alternative accelerated from IFASDA, fast-IFASDA, was also developed.

The method proposed by Yu Xiao et. al. [43] is geared towards restoring images corrupted by both Gaussian and impulse noise. It presents an approach whereby the relevant terms are respectively used for impulse denoising and partial representation in an undetermined image patch dictionary. The algorithm itself is comprised of three phases: in the first, the outlier candidates most susceptible to corruption by impulse noise are identified; the second involves the recovery of the image by way of dictionary learning on the free-outlier pixels; the third and final stage involves the use of an alternating minimization algorithm to find a solution to the proposed minimization energy function, resulting in an improved restoration based on the image recovered in the second phase. Results show that the proposed method fared better relative to other alternatives.

Lastly, Stefan et. al.'s [44] method establishes a framework for the development of nonspecific expressive image priors that describe the characteristics of natural scenes such that they can be applied in various machine-vision tasks. The method provides a practical way of studying the high-order Markov Random Field (MRF) models with possible functions covering a large area of pixels. The clique potentials are shaped using a Product-of-Experts framework, which itself uses the non-linear functions of numerous linear filter responses. Training data is used to map the values of all parameters – linear filters inclusive – in contrast to earlier MRF approaches. The proficiency of this Field-of-Experts model is confirmed by its potential application for both image inpainting and image denoising through a basic, imprecise inference scheme. The results that were obtained through this method are on such a level that they can compete with specialized techniques, even though the model itself was developed using a nonspecific image database that is not intended for specialized usage.

Chapter 3

JSM WITH NON-LINEAR FILTER

This chapter outlines an algorithm for use in image restoration based on the derivations provided in the previous chapters. All of the issues encountered when dealing with the three sub-problems outlined earlier (as seen in Section Appendix B.1) have been resolved in an efficient manner. In the first, several experiments are conducted in order to discover the best values for image restoration in the JSM algorithm. In the second, the method involves the addition of a Switching Median Filter (SMF) to the JSM and a Median Filter (MF) at the end of every iteration in the restoration process. In simple terms, the algorithm is essentially a hybrid denoising method that uses an improved SMF. The restoration of images is an important aspect of image processing that involves estimating a high-quality version of a given image of a considerably lower quality due to a lower resolution and the presence of noise. The main purpose served by the SMF here is to compare the given pixel values and the differences between the median values of pixels in the filtering window. Extensive experiments on image inpainting, image deblurring and mixed Gaussian plus “salt” and “pepper” noise removal applications validate the effectiveness of the proposed algorithm based on JSM.

3.1 Joint Statistical Modeling (JSM)

In addition to a previous mention in Section 2.4, further formulations of JSM can be found in Appendix B. So the Split Bergman Method can be found in Appendix A.

3.1.1 Image Inpainting

If we take Eq. B.1 (see Appendix B for u sub-problem) as a minimization problem of a strictly convex quadratic function, there is actually a closed form for u, which is written as:

$$u = (H^T H + \tilde{\mu} I \ )^{-1} . q, \tag{3.1}$$

where $q = H^T y + \mu 1(w + b) + \mu 2\ (x + c\ ), I$ is an identity matrix, and $\tilde{\mu} = \mu 1 + \mu 2$. Eq. 3.1 can be efficiently calculated for problems related to image inpainting and image deblurring. In regard to image inpainting, because the sub-sampling matrix H is in fact a binary matrix that can be generated using the subset of an identity matrix's rows, H satisfies $H^T H = I$. The application of the Sherman-Morrison-Woodbury (SMW) matrix inversion formula to Eq. 3.1 results in the following formulation:

$$u = \frac{1}{\tilde{\mu}} (I - \frac{1}{1 + \tilde{\mu}} H^T H\ ). q \tag{3.2}$$

Therefore, to efficiently calculate for u in Eq. 3.2 without calculating the matrix inverse in Eq. 3.1. Since $H^T H$ is equal to an identity matrix with a particular number of 0's in its diagonal, these 0's correspond to the locations of the pixels that are missing. As such, the cost associated with Eq. 3.2 is entirely equal to O(N).

Image inpainting is done by following steps:

- First, read the damaged image and the mask (text image) to fill it.
- Clear damaged area by killing or keeping the pixels by replacing it with either 0 or 1, respectively.

3.1.2 Salt-and-Pepper

Noise removal using a mixture of Gaussian and "salt" and "pepper" noise removal is considered to be a unique case of image inpainting, which is discussed in detail in the next section.

3.1.3 Image Deblurring

In regards to image deblurring, H represents a circular convolution. This convolution is factorized as:

$$H = U^{-1} D U, \tag{3.3}$$

where the 2D, DFT is denoted by the matrix U, which has the inverse U^{-1}, and H, which represents the convolution operator, has its DFT coefficient contained in the diagonal matrix D. As such:

$$(H^T H + \tilde{\mu} I)^{-1} = (U^{-1} D^* D U + \tilde{\mu} U^{-1} U)^{-1} = U^{-1} (|D|^2 + \tilde{\mu} I)^{-1} U, \tag{3.4}$$

where $(\cdot)^*$ denotes a complex conjugate and $|D|^2$ is the squared absolute values of entries in the diagonal matrix D. Due to the fact that $|D|^2 + \tilde{\mu} I$ is a diagonal, the cost of its inversion is $O(N)$; in practice, the products of U^{-1} and U can be implemented with $O(NlogN)$ using the FFT algorithm.

Algorithm 1. Image Deblurring

Step 1: We take the original image as the input image.
Step 2: Original image is then convoluted with PSF
Step 3: Apply Inverse FFT.
Step 4: Convoluted output is added with motion blur, uniform blur and Gaussian blur
Step 5: Apply either motion blur, uniform blur or Gaussian blur to get the blurred image.
Step 6: The output of the blurry image is then de-convoluted by JSM.
Step 7: After de-convolution, we get the deblurred image.

Algorithm 2s a description of JSM after adding a MF at the end of every iteration in the restoration process.

Algorithm 2 a complete description of proposed method using JSM

```
Input: y (observed image) and H (linear matrix operator)
Initialization: $\kappa = 0, u^{(0)} = \mathcal{Y}, b^{(0)} = c^{(0)} = \mathcal{w}^{(0)} = x^{(0)} = 0, \tau, \lambda, \mu 1, \mu 2;$
Repeat
Compute $u^{(k+1)} by\ Eq.(3.4)\ or\ Eq.(3.2);$
$\mathcal{P}^{(k)} = u^{(k+1)} - b^{(k)};\ \mathcal{Y} = \frac{\tau}{\mu 1};$
Compute $\mathcal{w}^{(k+1)}\ by\ FISTA;$
$\mathcal{r}^{(k+1)} = u^{(k+1)} - c^{(k+1)};\ a = \frac{\lambda}{\mu 2};$
Compute $x^{(k+1)} = by\ Eq.(B.3);$
$b^{(k+1)} = b^{(k)} - \left(u^{(k+1)} - \mathcal{w}^{(k+1)}\right);$
$c^{(k+1)} = c^{(k)} - \left(u^{(k+1)} - x^{(k+1)}\right);$
$M^{(k)} = Median\left\{u^{(k+1)}\right\}$
Until the highest possible iteration number has been reached.
Output: $u$ (the restored image)
```

Figure 3.1 shows a flow diagram of JSM where it can be seen that the input image is subjected to three types of noise:

(I) Blurred image.

(II) Inpainted image by text.

(III) Mixed Gaussian noise pulse “salt” and “pepper” noise.

A series of experiments (the output images) on blur removal and inpainting removal, and noise removal.

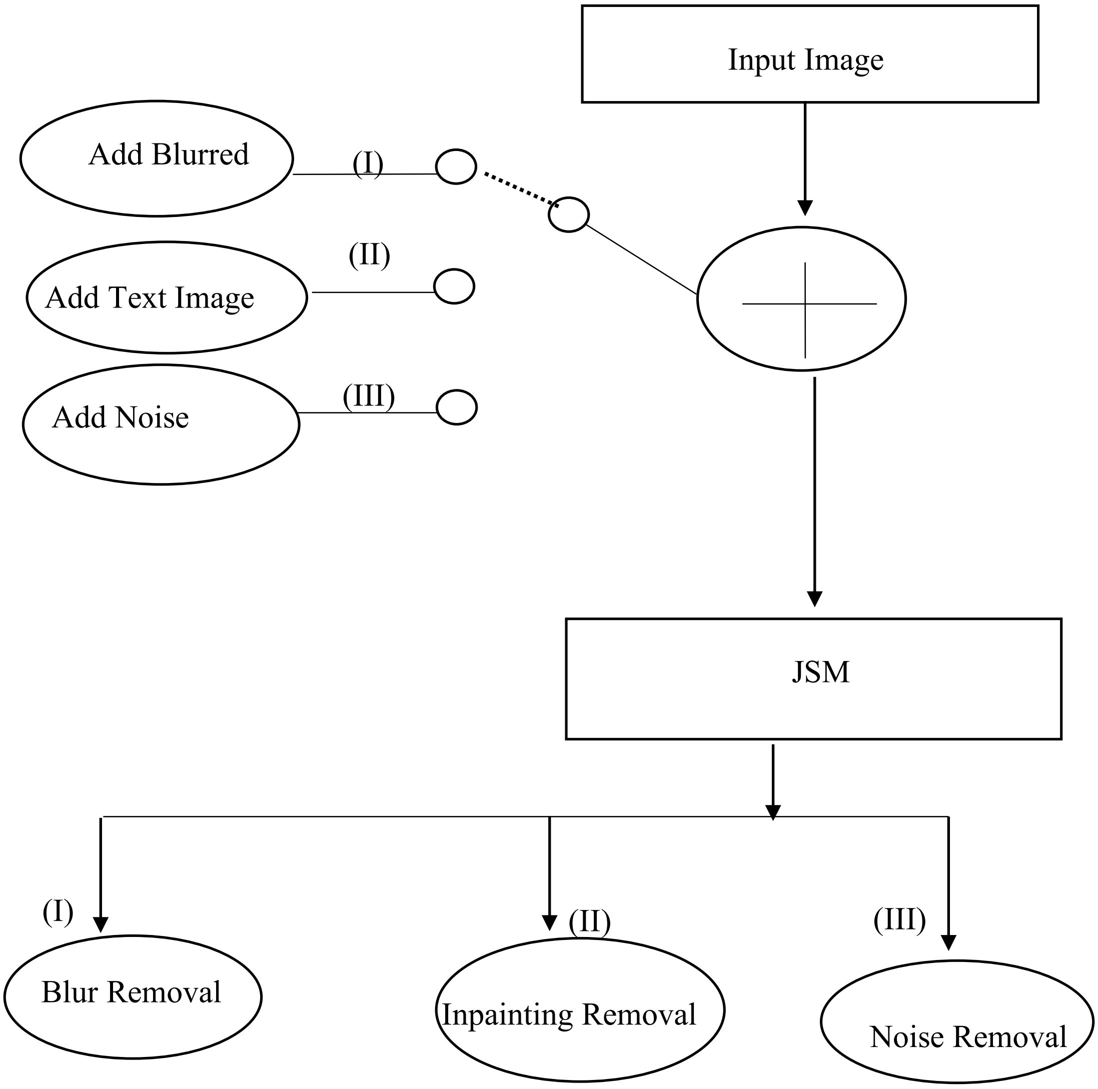


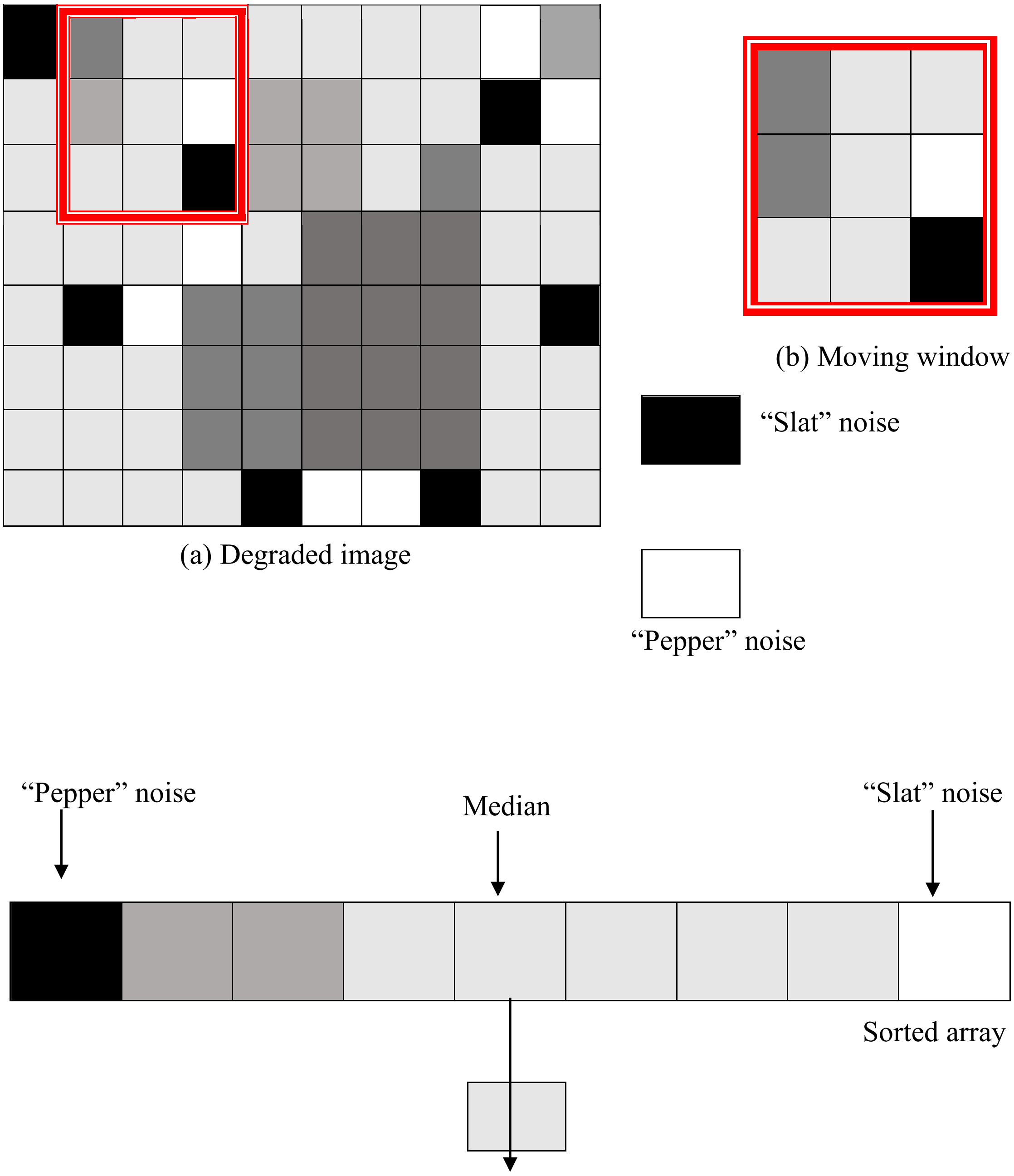
Figure 3.1: Flow Diagram of different images reconstruction process using JSM

3.2 Non-Linear Filtering

The primary purpose of a nonlinear filter is the location and removal of data considered to be noise. It is nonlinear because it individually evaluates data points to determine whether or not they are noise. Noisy data points are subsequently removed and replaced with an estimate derived from neighboring data points, while non-noisy data points are left unmodified. Linear filters, such as those used in high, low, and bandpasses, do not have the capacity to differentiate noise and thus modify all data points. Nonlinear filters also find occasional use in ridding data of brief wavelengths with high amplitude features. Filters of this sort are known as noise spike-rejection filters and are also useful for the removal of the geological features of short wavelengths. The process of image restoration involves the estimation of a clean, original image from a noisy/corrupted image; the image corruption may manifest in a number of ways, such as motion blur noise [19].

3.2.1 Median Filtering(MF)

One type of non-linear filtering is MF. This technique is renowned for its ability to preserve sharp edges while simultaneously removing impulsive types of noise. As an order statistics filter, it does not involve the replacement of pixel values with the average of the surrounding values, but with their median instead. This median is calculated by numerically sorting the values of the neighboring pixel and then replacing the noisy pixel with the middle value [53]. As Figure 3.2, shows the technique of an MF, (a) degraded image, (b) a matrix (3x3) which we selected from the degraded image. Then we convert all the 3x3 matrix elements into single row matrix ordering from smallest to highest as shown in part (c). The black color represents pepper noise while the white color represents salt noise. The other colors are pixels` between salt and pepper. In (c) we find median by calculating numerically and sorting the values of the neighboring pixels from black color to white color in the array and the replacing the noisy pixel with the middle value.



(a) Degraded image

(b) Moving window

(c) Filter output

Figure 3.2: Technique of MF

3.2.1.1 Switching Median Filter(SMF)

The SMF is characterized by a comparison of the difference between the median pixel value in the filtering window and the present value with a predetermined threshold in an attempt at determining whether or not an impulse is present. Only pixels found to have been subject to impulse noise are subsequently filtered. This method finds its basis in two schemes: a switching scheme, whereby only a fraction of pixels is filtered due to the use of an impulse detection scheme; and progressive methods, where a number of iterations are subject to both noise filtering and impulse detection. The method is primarily advantageous in that it can lead to better restoration results (particularly in extremely corrupted images) by properly detecting and filtering impulse pixels contained in large blotches of noise [52].

The main purpose of the SMF is to observe each and every pixel from the beginning till the end. The process begins with the classification of pixels into either of 3 categories: low-intensity, medium-intensity, and high-intensity pixels. In a 3x3 window, the pixels adjacent to the center pixel are evaluated. If the center pixel is outside the range of medium-intensity, then all the pixels are taken to be corrupt. Accurate boundary values are intrinsic to the determination of an accurate range of intensity. All the pixels in the noisy image undergo the same process to determine whether or not they are corrupted. This process involves the formation of a two-dimensional map with values 0 (uncorrupted pixel) and 1(corrupted pixel). To do this, two boundaries – B1 and B2 – are determined for each pixel being processed. The pixel is considered to be low-intensity when $0 < X(i, j) < B1$ medium-intensity when $B1 < X(i, j) < B2$ and high-intensity when $B2 < X(i, j) < 255$. There are two iterations contained in the algorithm; the first involves determining, by increasing the size of the window, whether or not an uncorrupted pixel is present and if none is found, the next iteration can be omitted.

Consider the following matrix below as an example (see Algorithm 3):

The steps necessary in determining whether or not impulse noise is present in an M×N sized image with an **8**-bit gray-scale pixel resolution are outlined in Figure 3.3:

Step 1) As shown in the image below, a **3x3** two-dimensional filtering window is first superimposed on a contaminated image.

161	162	159	163	63
167	255	0	255	255
164	255	255	255	255
165	0	255	255	255
166	255	159	255	167

Figure 3.3 Matrix (5*5)

Step2) The pixels within the window are then numerically arranged.

0 159 162 163 **255** 255 255 255 255

Step 3) Here, we determine the minimum, maximum, and median values in the window. In the example above, these are **0, 255** and **255**.

Step 4) Central pixels with a value between the maximum and minimum are considered to be uncorrupted and are left unmodified. In cases where the pixel is not between the maximum and minimum values, the relevant pixel is considered to be corrupted such as in the present example where the central pixel **255** is also the maximum.

Step 5) Here, the median value in the window is used in replacing the corrupted central pixel provided the median itself is not an impulse. If it is an impulse, then the immediate top neighboring pixel $A_{i-1,j}$ is used to replace the central pixel in the filtering window. The present example is an instance of an impulse median and so the pixel is replaced by the top value: **159**.

161	162	159	163	63
167	255	159	255	255
164	255	255	255	255
165	0	255	255	255
166	255	159	255	167

Figure 3.4: Matrix (5*5)

Figure 3.4 shows how the window is subsequently relocated to cover a new collection of pixel values for which the relevant pixel is in the center. This process is repeated until all of the image's pixels have been processed. The following conditions forms the basis for the detection and filtering of impulse noise:

Algorithm 3. Switching Median Filter

```
if $A_{min}$ <$A_{i,j} < A_{max}$
        {$A_{i,j}$ $is\ a\ noiseless\ pixel$; no filtering is performed on $A_{i,j}$ }
else
        {$A_{i,j}$ $is\ a\ noisy\ pixel$; determine the median value}
        if median $\neq 0$ $and\ median\ \neq 255$
                {Median filter is performed on $A_{i,j}$ }
                                $A_{i,j} = A_{med}$
        else
                {Median itself is noisy}
                                $A_{i,j} = A_{i-1,j}$
        end;
end;
```

where in algorithm 3, $A_{i,j}$ is the intensity of central pixel inside the filtering window, A_{min}, A_{max} and A_{med} are the minimum, maximum and median pixel value in filtering window of the noisy image. $A_{i-1,j}$ is the intensity of the already processed immediate top neighboring pixel.

In processing the border pixels, the initial and final columns get duplicated, respectively, at the front and back of the image matrix. In a similar manner, the initial and final rows are also duplicated at the topmost and bottommost parts of the image matrix. The processing of the pixels in the first row uses the algorithm outlined above, except for Step 5 – an impulse median value is to be replaced by the nearest untouched neighboring pixel in the filtering window.

Figure 3.5 shows a flowchart for the SMF. The steps involved in using a SMF are as follows:

Step-1: In a large window with x (i, j) at its center, the current pixel in the image should be the center pixel.

Step-2: The pixels are numerically arranged and stored in an array A. The median is then determined and the result is stored in M.

Step-3: For every pair of adjacent pixels within the array *A,* the intensity difference is calculated and the result is stored in the difference vector *Ad*.

Step-4: Find the pixels from *Ad* that corresponds to the maximum differences in the intervals of [0, *M*] and [*M*, 255]. Set these pixel's intensities as the decision boundaries B1 and B*2* respectively.

Step-5: If the processing pixel belongs to the middle cluster then it is classified as uncorrupted and the process stops. Otherwise, it must go for the second iteration, which will be invoked as follows.

Step-6: Steps 2-4 are repeated in an imposed 3x3 window centered on the relevant pixel.

Step-7: The current pixel is uncorrupted if it belongs to the middle cluster and is corrupted otherwise.

Step-8: Based on this algorithm we are updating the detection map. If the processing pixel is uncorrupted the detection map is updated with “0” otherwise detection map is updated with “1”.

Step-9: Here, we arrive at the restored Image.

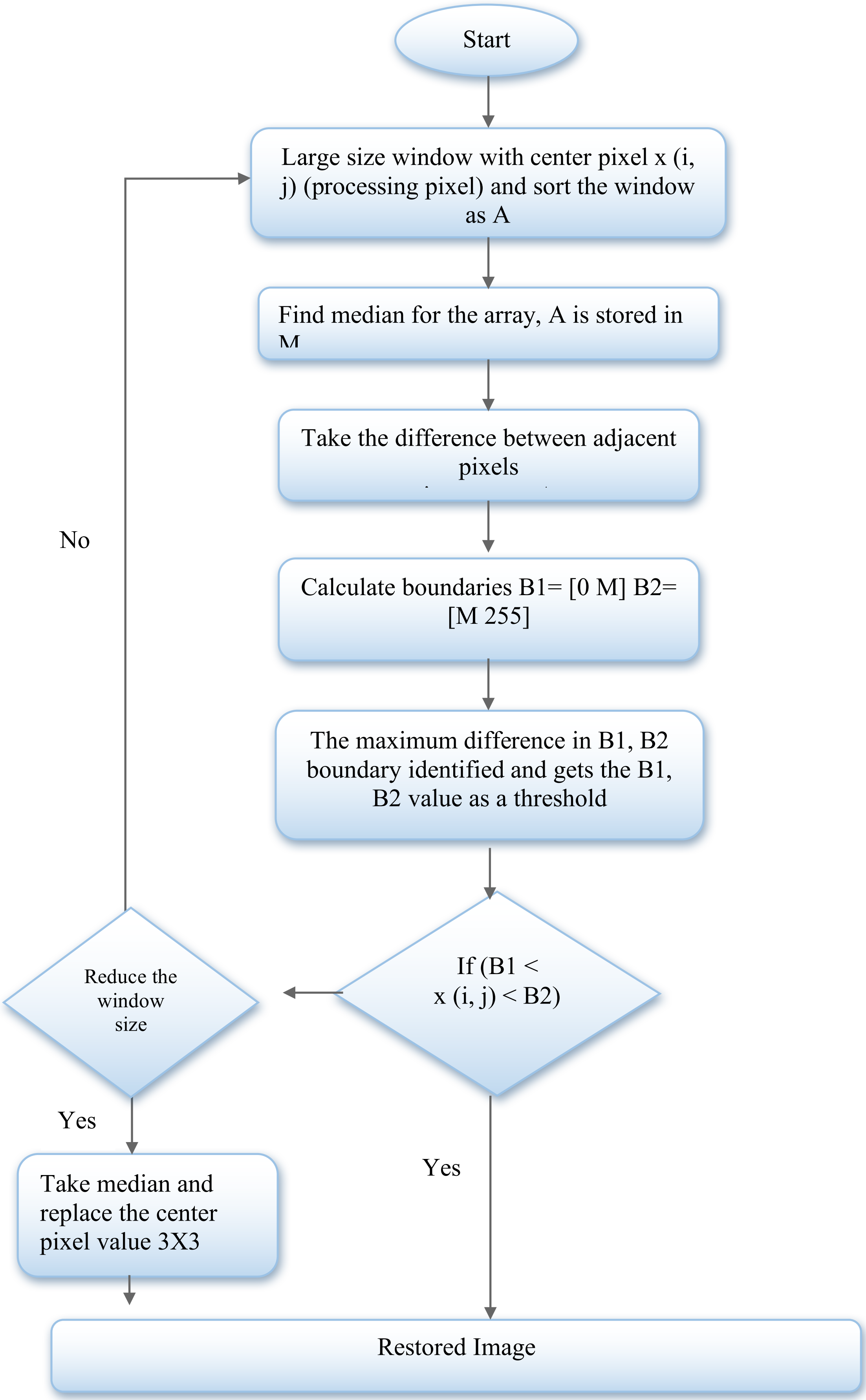

Start
Large size window with center pixel x (i, j) (processing pixel) and sort the window as A
Find median for the array, A is stored in M
Take the difference between adjacent pixels
Calculate boundaries B1= [0 M] B2= [M 255]
The maximum difference in B1, B2 boundary identified and gets the B1, B2 value as a threshold
If (B1 < x (i, j) < B2)
Reduce the window size
No
Yes
Take median and replace the center pixel value 3X3
Yes
Restored Image

Figure 3.5: Flowchart of SMF

Figure 3.6 shows the block diagram for JSM with a MF. The proposed method is a variation of JSM with the addition of a SMF and a MF at the end of every iteration in the restoration process.

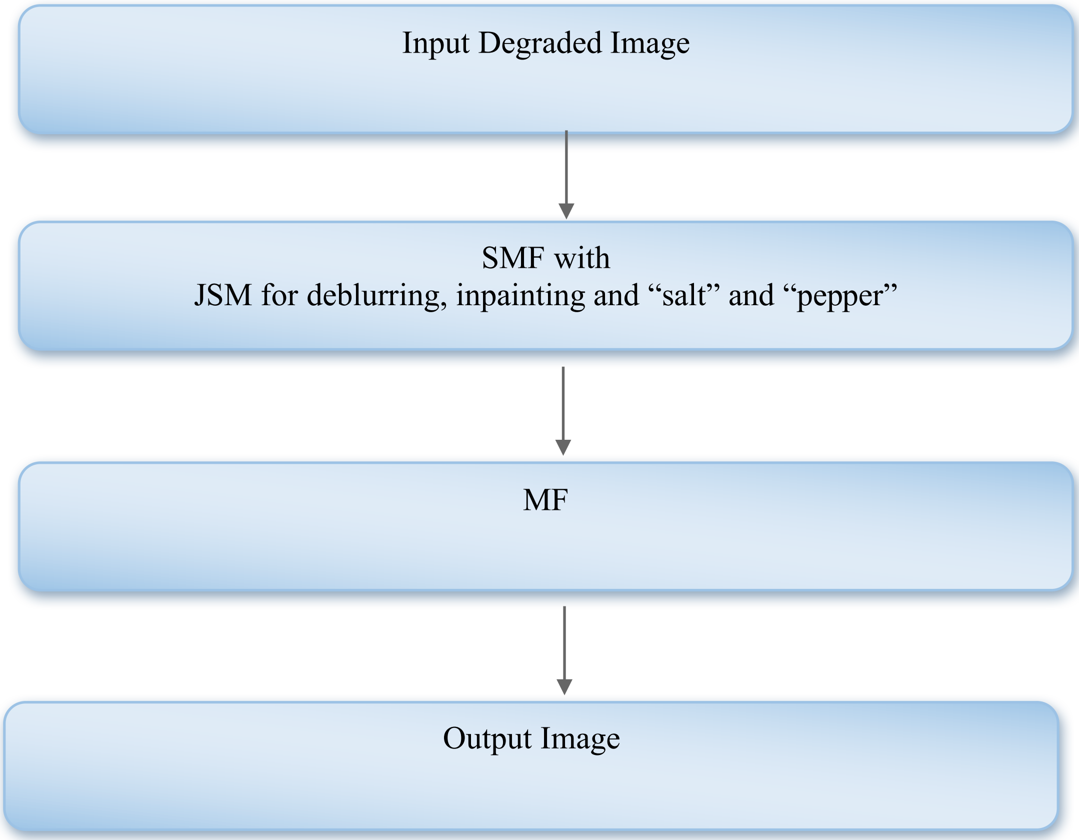


Figure 3.6: Block diagram for the JSM with MF

3.3 Mean Squared Error (MSE)

MSE is an error measurement technique; it is used for a pixel-by-pixel computation of the mean square error of the test image in comparison to the original [54] and is usually mathematically written as:

$$\text{MSE} = \frac{1}{MN}\sum_{X=0}^{M-1}\sum_{Y=0}^{N-1}[\mathrm{f}(x,y)-\mathrm{g}(x,y)]^2 \tag{3.5}$$

where *g*(*x,y*) and *f*(*x,y*) are the distorted and reference images of pixels with a size MxN, respectively. This metric is advantageous in that it is simple; it does, however, correlate poorly with subjective results.

3.3.1 Peak Signal to Noise Ratio (PSNR)

This method similarly involves a pixel-by-pixel comparison of the reference and distorted images and is mathematically written as [54]:

$$\text{PSNR=10log}\left(\frac{(2^B-1)^2}{\frac{1}{MN}\sum_{X=0}^{M-1}\sum_{Y=0}^{N-1}[\text{f}(x,y)-\text{g}(x,y)]^2}\right) \tag{3.6}$$

Eq. (3.6) can alternatively be denoted as:

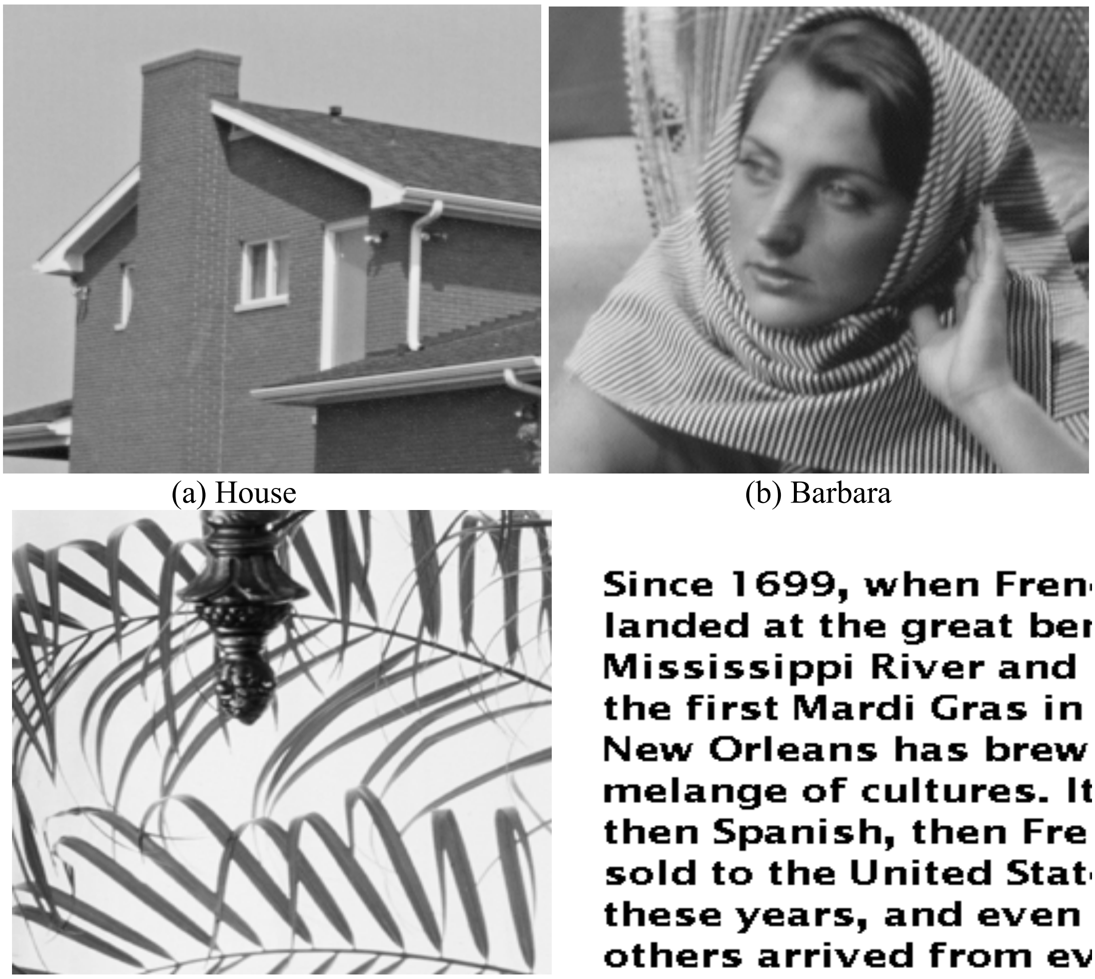
$$\text{PSNR} = \text{10log}\left(\frac{(2^B-1)^2}{MSE}\right)\text{ dB} \tag{3.7}$$

Chapter 4

RESULTS AND DISCUSSION

4.1 Results and Discussion

The primary objective of this thesis has been the use of high-precision image restoration in the production of a high-quality image. This process is based on the statistical characterization of the nonlocal self-similarity and local smoothness of natural images. Here, we present the results of image restoration using the proposed algorithm in comparison to pure JSM. The algorithms mentioned in Chapter 3 are applied to image deblurring, image inpainting, and mixed Gaussian plus “salt” and “pepper” noise removal. All the experiments are performed in Matlab R2016a on a MacBook Pro (Retina, 13-inch, Early 2015), processor 2.9 GHz Intel Core i5, memory 8 GB 1867 MHz DDR3, and operating system. In our implementation, if not specially stated, the size of each block is set at 8x8 with a 4-pixel separation between adjoining blocks; the size of training window for searching matched blocks, i.e., L×L is set to be 40×40, and the number of best-matched blocks, i.e., c is set to be 10. Thus, the relationship between N and K is K =40N. The PSNR (measured in dB) is used for evaluating the quality of the post-reconstruction image. The cover images and text mask (grayscale images) are: "House", "Barbara", "Leaves" and "TextMask", selected for experiments; the size of each image is set to 256 x 256 pixels as shown in Figure 4.1.



(a) House (b) Barbara

Since 1699, when Fren
landed at the great bel
Mississippi River and
the first Mardi Gras in
New Orleans has brew
melange of cultures. It
then Spanish, then Fre
sold to the United Stat
these years, and even
others arrived from ev

(c) Leaves (e) TextMask

Figure 4.1: Cover-Images and Text Mask

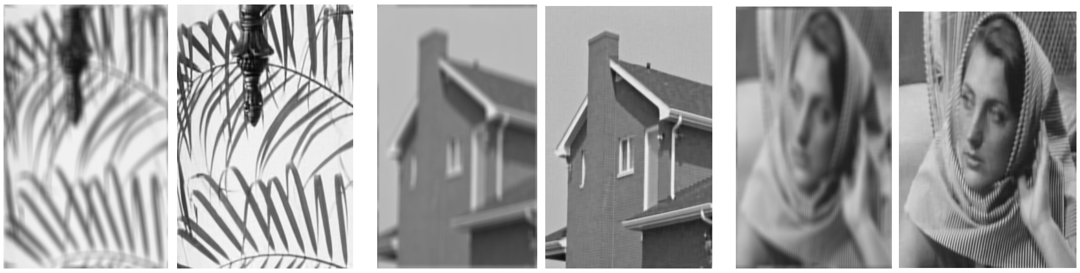
4.1.1 Image Deblurring

The original images subject to image deblurring are initially blurred using a blur kernel, to which Gaussian noise is added using standard deviation. The simulation utilizes three blur kernels: a motion blur kernel, a Gaussian blur kernel, and a 9x9 uniform kernel. The image deblurring results get from the proposed method are then compared to JSM (shown in Table 4.1). A visual comparison of the blurred and deblurred images using the proposed method is provided in Figures 4.2, 4.3 and 4.4.

Figure 4.2 shows the effect of a uniform blur for the three test images, these are Leaves, House and Barbara. In Figure 4.2 (**a**) the original image Leaves is blurred with a (9*9) uniform blur, then the deblurring result is demonstrated using the proposed method with PSNR **32.77dB**. Figure 4.2 (**b**) shows the effect of blurring for the image House. The original image made noisily and blurred with a (9*9) uniform blur. Here the deblurring result using the proposed method is achieved with PSNR **39.74dB**. Finally, in Figure 4.2 (**c**) the same process is applied to image Barbara. The deblurring result using the proposed method is achieved with PSNR **33.29dB**.

Figure 4.3 shows the effect of Gaussian blur of size 25 pixels, with the standard deviation sigma 1.6 for the three test images, there are Leaves, House and Barbara. In Figure 4.3 (**a**) the original image Leaves is blurred with a Gaussian blur, then the deblurring result is demonstrated using the proposed method with PSNR **33.54dB**. Figure 4.2 (**b**) shows effect of blurring for the image House. The original image made noisy and blurred with a Gaussian blur. Here the deblurring result using the proposed method is achieved with PSNR **37.15dB**. Finally, in Figure 4.2 (**c**) the same process is applied to image Barbara. The deblurring result using the proposed method is achieved with PSNR **31.25dB**.

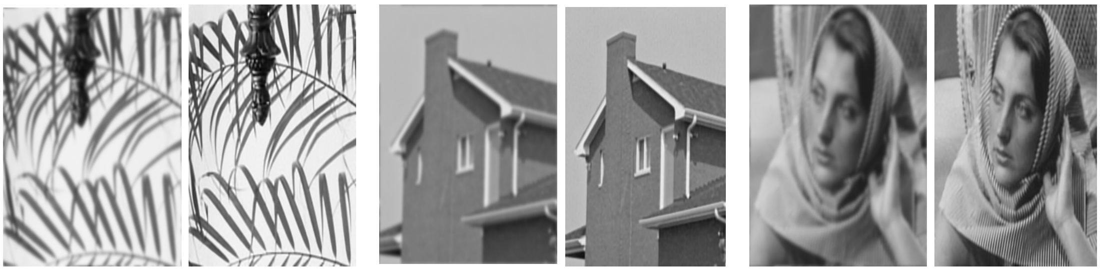
Figure 4.4 shows the effect of motion blur of length 20 pixels and an angle of 45 degrees for the three test images, there are Leaves, House and Barbara. In Figure 4.4 (**a**) the original image Leaves is blurred with a motion blur, then the deblurring result is demonstrated using the proposed method with PSNR **38.26dB**. Figure 4.4 (**b**) shows effect of blurring for the image House. The original image made noisy and blurred with a motion blur. Here the deblurring result using the proposed method is achieved with PSNR **39.04dB**. Finally, in Figure 4.4 (**c**) the same process is applied to image Barbara. The deblurring result using the proposed method is achieved with PSNR **41.72dB**.



i ii i ii i ii

(a) **(b)** **(c)**

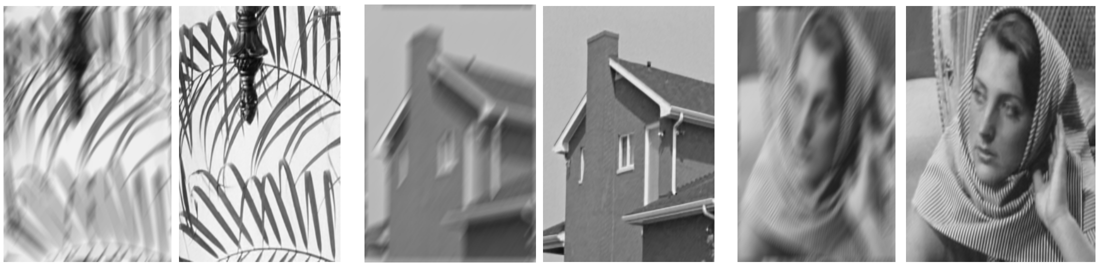
Figure 4.2: Image deblurring with uniform blur



i ii i ii i ii

(a) **(b)** **(c)**

Figure 4.3: Image deblurring with Gaussian blur



i ii i ii i ii

(a) **(b)** **(c)**

Figure 4.4: Image deblurring with motion blur

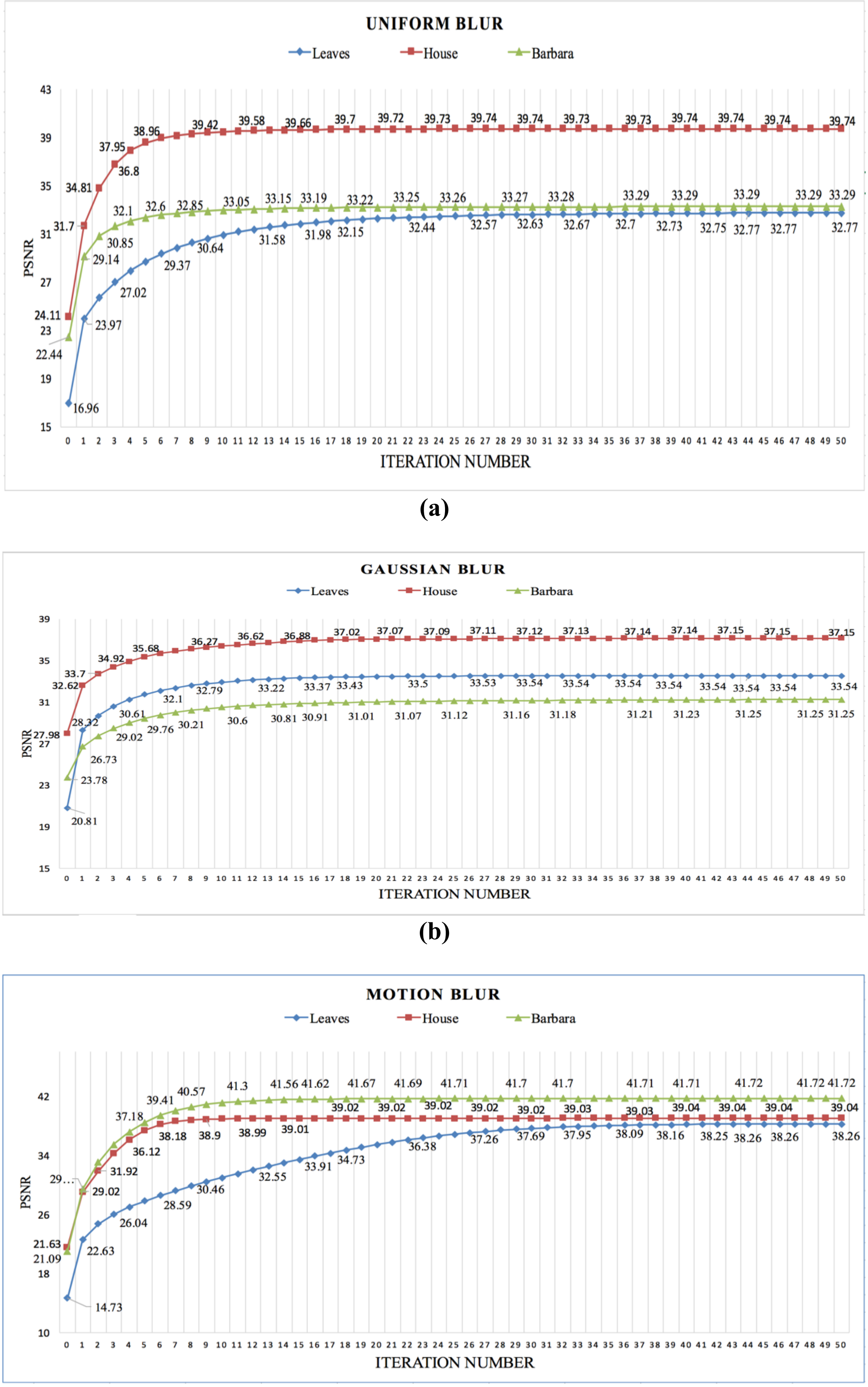
The graph line in Figure 4.5 illustrates the relationship between the PSNR and number of iterations used in deblurring for the three test images and thus. The plots show the evolution of the PSNR in relation to the iteration numbers for various initializations of the test images. It is evident from the plots that the PSNR curve monotonically increases and tends towards convergence with each additional iteration, thus proving the convergence of the proposed method. Further details are provided below.

Figure 4.5 (a) shows the three test images for image deblurring. The original images are blurred by a 9*9 uniform blur and sigma **0.5**, the deblurring image has been obtained by a minimum of **30** to **50** iterations. For image House, the initial PSNR is **24.11dB**, which increases to **31.70dB** in the first iteration. When the iteration number reaches **26** iterations, the high PNSR value is achieved to **39.74dB**. Then for image Leaves, the initial PSNR is **16.96dB**, then in the first iteration, the PSNR increases to **23.97dB**. When the iteration number reaches **44** iterations, the high PNSR value is achieved to **32.77dB**. Finally, the initial PSNR of image Barbara is **22.44dB**, then in the first iteration, the PSNR is increased to **29.14dB**. When the iteration number reaches **35** iterations, the high PNSR values is achieved to **33.29dB**. Overall, the high PSNR values gave us a good result.

Figure 4.5 (b) shows three tested images for image deblurring. The original images are blurred with a Gaussian blur of size **25** pixels with a standard deviation sigma of **1.6** and sigma **0.5**; the deblurring image has been achieved in a minimum of **30** to **50** iterations. For image House, the PSNR is **27.98dB**; after the first iteration, this increases to **32.62dB**. When the iteration number reached **40** iterations, the highest PNSR value was achieved at **37.15dB**. Then for image Leaves, the initial PSNR is **20.81dB**, then in the first iteration, the PSNR increases to **28.32dB**. When the iteration number reaches **28** iterations, the highest PNSR value is achieved at **33.54dB**. Finally, the initial PSNR of image Barbara is **23.78dB**, then in the first iteration, the PSNR is increased to **26.73dB**. When the iteration number reaches **43** iterations, the high PNSR value is achieved to **31.25dB**.

Figure 4.5 (c) shows three tested images for image deblurring. The original image is blurred with a motion blur of length **20** pixels and an angle of **45** degrees, and sigma **0.5**; the deblurring image has been achieved in a minimum of **30** to **50** iterations. For image House, the initial PSNR is **21.63dB**, then in the first iteration, the PSNR is increased to **29.02dB**. When the iteration number reached **37** iterations, the high PNSR value is achieved at **39.04dB**. Then for image Leaves, the initial PSNR is **14.73dB**, then in the first iteration, the PSNR is increased to **22.63dB**. When the iteration number reaches **44** iterations, the high PNSR value is achieved to **38.26dB**. Finally, the initial PSNR of image Barbara is **21.09dB**, then in the first iteration, the PSNR is increased to **29.02dB**. When the iteration number reaches **41** iterations, the high PNSR values is achieved to **41.72dB**. Here, the high PSNR values also gave us a good result.

In summary, the convergence of the algorithm simultaneously simplifies the execution of the proposed algorithm, which is only intended to run for a predetermined number of iterations. Additionally, the higher the quality of the initialization result, the fewer the number of iterations requires to reach convergence (see Figure 4.5).



(a)

(b)

(c)

Figure 4.5: Verification of the convergence and robustness of the proposed algorithm. In the cases of image deblurring; (a) uniform blur; (b) Gaussian blur; (c) motion blur.

As shown in Table 4.1. First, in the case of the uniform blur table for image House, the JSM was checked as **37.73dB** while the proposed method result is **39.74dB**. In percentage terms, the PSNR is improved by **5.33%**. For the image Leaves, the JSM was checked as **31.61dB**; the proposed method result is **32.77dB**. The calculated percentage PSNR improvement is **3.67%**. For the image Barbara, the JSM was checked as **29.65dB**. The proposed method result is **33.29dB**. The PSNR experienced an improvement of as much as **12.28%**.

Second, in the case Gaussian blur table for image House, the JSM was checked as **36.68dB**. The proposed method result is **37.15dB**. The calculated percentage PSNR improvement stood at **1.28%**. For the image Leaves, the JSM was checked as **32.18dB**. The proposed method result is **33.54dB**. The calculated percentage PSNR improved by **4.23%**. For the image Barbara, the JSM was checked as **28.66dB**. The proposed method result is **31.25dB**. The calculated percentage PSNR improved by as much as **9.04%**.

Finally, for the motion blur table for image House, the JSM was checked as **37.40dB**. The proposed method result is **39.04dB**. Calculating the percentage PSNR difference reveals an improvement of **4.39%**. For the image Leaves, the JSM was checked as **33.95dB**. The proposed method result is **38.26dB**. The calculated percentage PSNR improvement stood as high as **12.70%**. For the image Barbara, the JSM was checked as **34.25dB**. The proposed method result is **41.72dB**. The calculated percentage PSNR improved by an impressive **21.81%**.

Summarily, it is evident that the proposed method significantly improves the results of image deblurring when compared to JSM as the PSNR value of the proposed method is considerably higher than in JSM. So the results of PSNR value is different for different images or for different formatted images since it is related to image quality and noise in the images.

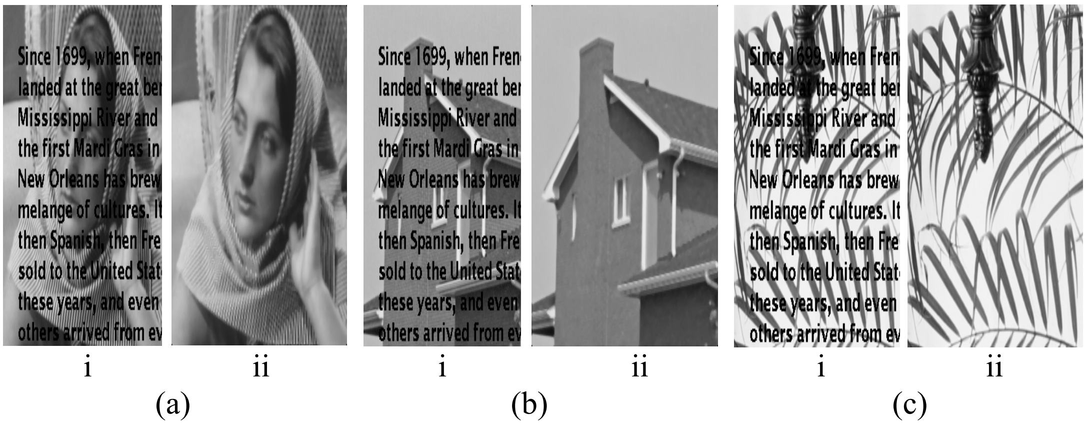
Table 4.1: PSNR for image deblurring (JSM vs proposed method)

Images	House	Leaves	Barbara
9*9 uniform Blur			
JSM (in dB)	37.73	31.61	29.65
Proposed Method (In dB)	39.74	32.77	33.29
Relative Improvement %	5.33%	3.67%	12.28%
Gaussian Blur			
JSM (in dB)	36.68	32.18	28.66
Proposed Method (in dB)	37.15	33.54	31.25
Relative Improvement %	1.28%	4.23%	9.04%
motion Blur			
JSM (in dB)	37.40	33.95	34.25
Proposed Method (in dB)	39.04	38.26	41.72
Relative Improvement %	4.39%	12.70%	21.81%

4.1.2 Image Inpainting

The primary purpose of text removal is the recovery of regions degraded by the pixels of a text to arrive at the original image. The simulation in this thesis utilizes a number of aspects, including text removal. The results of the proposed inpainting method are also compared to JSM (see Table 4.2).

Figure 4.6 shows the visual quality comparison of text removal for image inpainting on three tested images, Leaves, House and Barbara. In Figure 4.6 (**a**) shows the image Barbara as a degraded image with a text mask and the PSNR of image Barbara after the added text is **12.93 dB**; then the inpainting result is demonstrated using the proposed method with PSNR **48.38 dB**. Figure 4.6 (**b**) shows the image House as a degraded image with a text mask and the PSNR of image House after the added text is **12.91 dB**; then the inpainting result is demonstrated using the proposed method with PSNR **46.67 dB**. Finally, Figure 4.6 (**c**) the same process is applied to image Leaves with PSNR **9.70 dB**. The inpainting result using the proposed method is achieved with PSNR **39.92 dB**.


i ii i ii i ii

(a) (b) (c)

Figure 4.6: Visual quality comparison of text removal for image inpainting

The graph line in Figure 4.7 illustrates the relationship between the PSNR and number of iterations used for inpainting the three test images House, Leaves and Barbara in a single plot. The plot shows the evolution of the PSNR in relation to the iteration numbers for various initializations of the test images. It is evident from the plots that the PSNR curve monotonically increases and tends towards convergence with each additional iteration, thus proving the convergence of the proposed method.

In this case of image inpainting, a text mask is added to the original image and the purpose of text removal is to extrapolate the original images from their degraded versions by removing the text region. It achieves image inpainting in a minimum of **40** to **50** iterations. The PSNR values of the proposed method are increased after **50** iterations. For image House, the initial PSNR is **35.29dB**, then in the first iteration, the PSNR increases to **40.47dB**. When the iteration number reached **41** iterations, the high PNSR values is achieved to **46.67dB**. Image Leaves, the initial PSNR is **27.90dB**, then in the first iteration, the PSNR is increased to **31.41dB**. When the iteration number reaches **42** iterations, the high PNSR values is achieved to **39.92dB**. Finally, the initial PSNR of image Barbara is **28.86dB**, then in the first iteration, the PSNR increases to **33.61dB**. When the iteration number reaches **42** iterations, the high PNSR values is achieved to **48.38dB**. The high PSNR values give us the best results.

In summary, the convergence of the algorithm simultaneously simplifies the execution of the proposed algorithm, which itself is only intended to run for a predetermined number of iterations. Additionally, the higher the quality of the initialization result, the fewer the number of iterations requires to reach convergence.

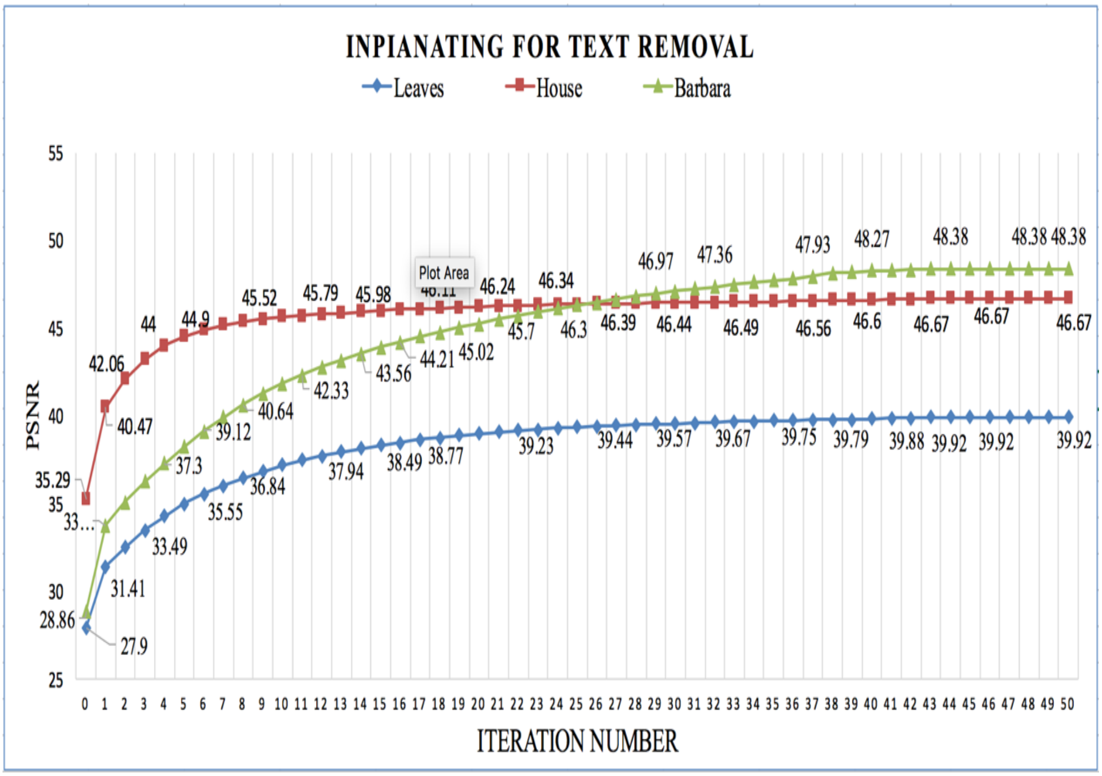


Figure 4.7: Convergence of inpainting process using the proposed method

Table 4.2 below shows the comparison for the inpainting images. Here, we compare the proposed inpainting method to JSM in the case of the three test images.

First, for the image house, the JSM was checked as **41.91dB**. The proposed method result is **46.67dB**. Calculating the percentage PSNR shows an improvement of **11.36%**. Second, for the Barbara image, the JSM was checked as **37.99dB**. The proposed method result is **48.38dB**. The calculated percentage PSNR improved by as much as **27.35%**. Finally, for the Leaves image, the JSM was checked as **34.04dB**. The proposed method result is **39.92dB**. The calculated percentage PSNR improved by **17.27%**.

In summary, it can clearly be seen that the proposed method is an improvement from the JSM as the PSNR value is significantly increased when compared to the JSMs of image inpainting. So the results of PSNR value is different for different images or for different formatted images since it is related to image quality and noise in the images.

Table 4.2: PSNR for image inpainting (JSM vs Proposed Method)

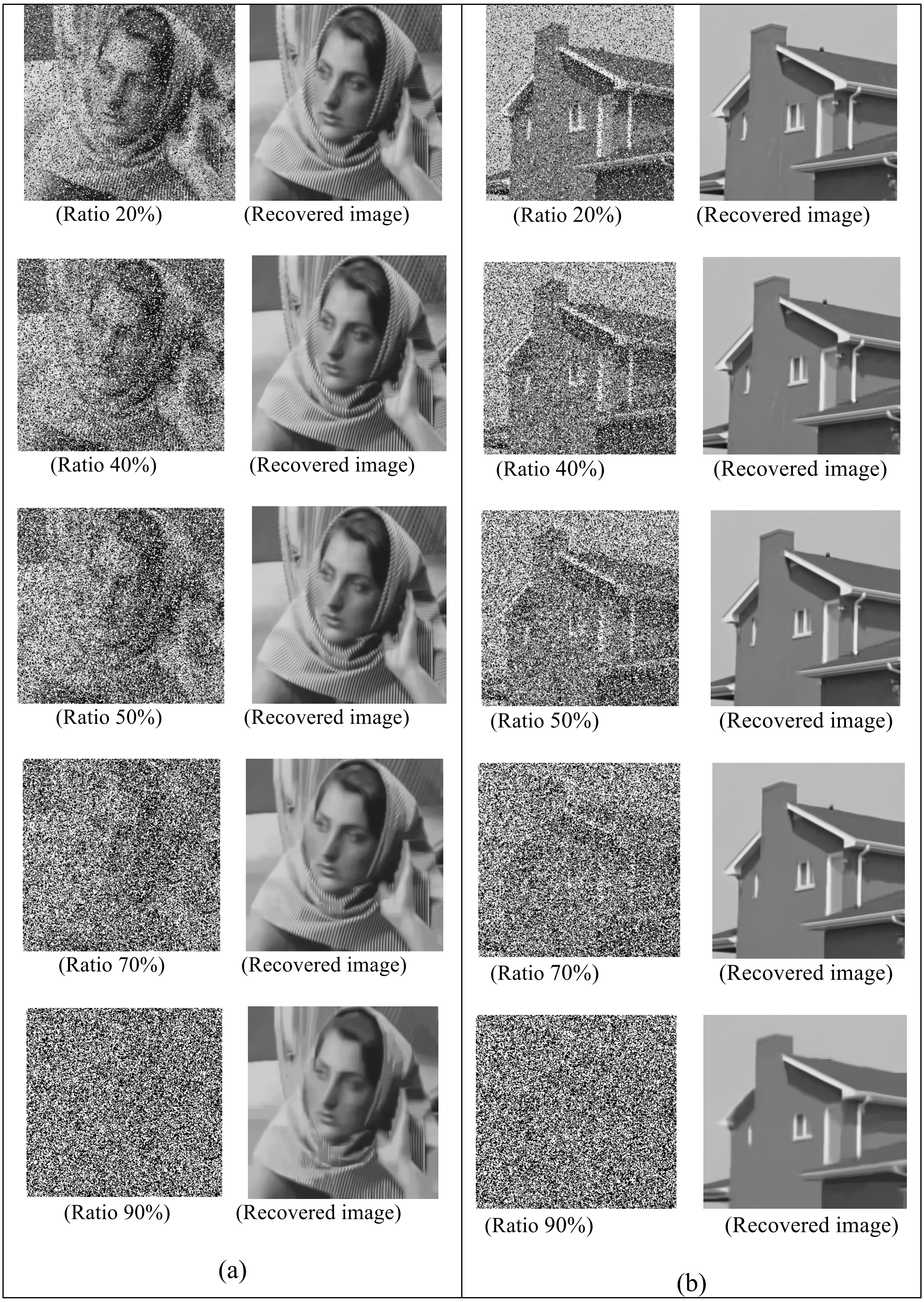
Images	House	Barbara	Leaves
JSM (in dB)	41.91	37.99	34.04
Proposed Method (in dB)	46.67	48.38	39.92
Relative Improvement %	11.36%	27.35%	17.27%

4.1.3 Mixed Gaussian plus Salt-and-Pepper Noise Removal

The images used in the simulations here are corrupted with Gaussian noise using standard deviation sigma and a salt-and-pepper noise density level *r*. Conventional methods for image denoising, capable only of handling a single kind of noise, are ineffective in the case of mixed Gaussian plus impulse noise due to the distinctiveness of the two degradation processes involved in this kind of noise.

A series of experiments are conducted on the two gray test images (see Figure 4.8), where the standard variance of Gaussian noise equals **10** and the noise density level *r* is in variations from **20%, 40%, 50% 70% and 90%.** Some visual results of the recovered images using the proposed method Are provided in Figure 4.8, including a visual quality comparison of mixed Gaussian plus salt-and-pepper impulse noise removal on the images House and Barbara. First, for the image Barbara, the PSNR for the noisy image corrupted by Gaussian plus “salt” and “pepper” impulse noise with ratio equal **20%** stood at **12.33 dB** while that of the denoised result using the proposed method was **38.10dB**; if the ratio equal **40%,** the PSNR reduces for both the original **9.37 dB** and the denoised result using the proposed method **36.39 dB**; if the ratio equal **50%,** PSNR reduces even further **8.40 dB** and the denoised result gotten through the proposed method has a PSNR of **34.74 dB**; the original PSNR and the denoised result using the proposed method are **6.09 dB** and **29.63 dB** if the ratio equal **70%**; and if the ratio equal **90%,** the PSNR is the lowest at **5.82 dB** and **22.61dB** for the original and the denoised result using the proposed method respectively, as shown in Figure 4.8 (a).

Second, the same process is applied for House, if ratio equal **20%,** the PSNR is **12.43 dB** and the denoised result using the proposed method has PSNR **37.30 dB**. Increasing the noise ratio reduces the PSNR and so when ratio equal **40%,** original PSNR is **9.46 dB** and **36.30 dB** for the denoised result from the proposed method; if the ratio equal **50%,** original PSNR **8.50 dB** and **35.44 dB** for the denoised result using the proposed method; if the ratio equal **70%,** original PSNR **7.01 dB** and that of the denoised result using the proposed method is **33.42 dB**; if the ratio equal **90%,** PSNR is the lowest for the original **5.93 dB** and the denoised result from the proposed method **27.84 dB**, as shown in Figure 4.8 (b).



(Ratio 20%) (Recovered image)

(Ratio 40%) (Recovered image)

(Ratio 50%) (Recovered image)

(Ratio 70%) (Recovered image)

(Ratio 90%) (Recovered image)

(a)

(Ratio 20%) (Recovered image)

(Ratio 40%) (Recovered image)

(Ratio 50%) (Recovered image)

(Ratio 70%) (Recovered image)

(Ratio 90%) (Recovered image)

(b)

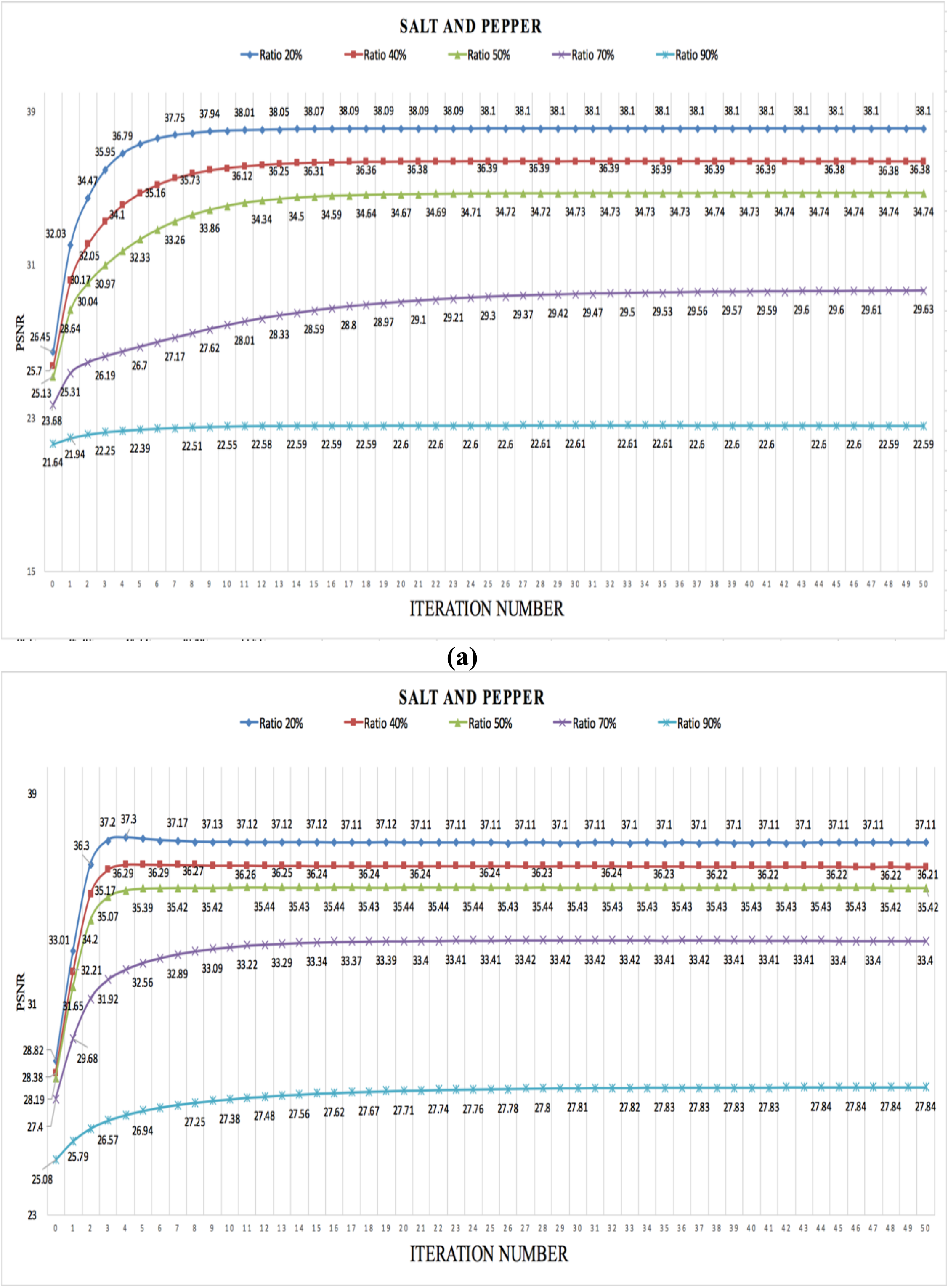
Figure 4.8: Visual quality comparison of mixed Gaussian plus salt-and-peppers impulse noise removal on images (a) Barbara and (b) House

The graph line in Figure 4.9 clearly shows the mixed Gaussian plus salt-and-pepper impulse noise removal for two images (a) Barbara and (b) House. The plots show the evolution of the PSNR in relation to the iteration numbers for various initializations of the test images. It is evident from the plots that the PSNR curve monotonically increases and tends towards convergence with each additional iteration, thus proving the convergence of the proposed method.

In this case, the original image will be corrupted by Gaussian noise with sigma and "salt-and-pepper" noise density level ratios are set to **20%,40%,50%,70%** or **90%** and the sigma of Gaussian noise equals **10**. Consequently, it achieves the Gaussian plus salt-and-pepper impulse noise removal in a minimum of **30** to **50** iterations.

First, as shown in the Figure 4.9 (a), the initial PSNR values of the image Barbara at ratios **20%,40%,50%,70%, and 90%** are **26.45, 25.70, 25.13, 23.68** and **21.64**dB, respectively. We then set the pre-specified upper bound as 50 for all noise levels. When the ratio is equal to **20%**, the high PNSR value is achieved at **38.10dB** after **23** iterations; when the ratio is equal to **40%**, the high PNSR value is equal to **36.39dB** after **24** iterations; when the ratio is equal to **50%**, the high PNSR value is achieved at **34.74dB** after **37** iterations; when the ratio is equal to **70%**, the high PNSR value is achieved at **29.63dB** after **45** iterations; and when the ratio is equal to **90%**, the high PNSR value is achieved at **21.61dB** after **26** iterations.

Second, as shown in Figure 4.9 (b), the initial PSNR values of the image House at ratios **20%,40%,50%,70%, and 90%** are **28.82, 28.38, 28.19, 27.40 and 25.08**dB. When the ratio is equal to **20%**, the high PNSR value is achieved at **37.30dB** after **3** iterations; when the ratio is equal to **40%**, the high PNSR value is achieved at **36.30dB** after **6** iterations; when the ratio is equal to **50%**, the high PNSR value is achieved at **35.44dB** after **11** iterations; when the ratio is equal **70%,** the high PNSR value is achieved at **33.42dB** after **25** iterations; and when the ratio is equal to **90%,** the high PNSR value is achieved at **27.84dB** after **40** iterations.



(a)

(b)

Figure 4.9: Convergence of noise removal process using the proposed method

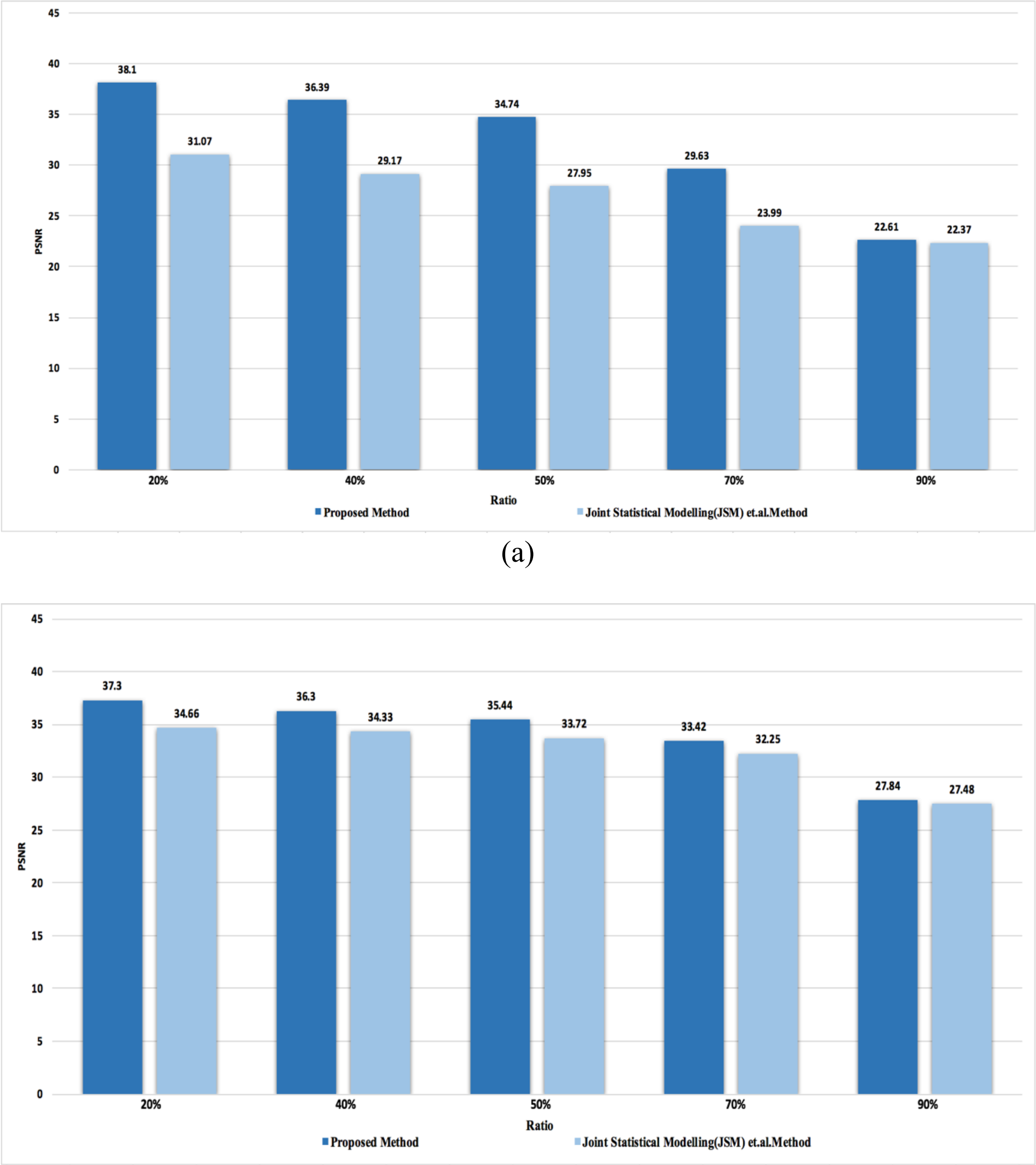
Summarily, the convergence of the algorithm simultaneously simplifies the execution of the proposed algorithm, which itself is only intended to run for a predetermined number of iterations. Additionally, the higher the quality of the initialization result, the fewer the number of iterations requires to reach convergence.

Figure 4.10 provides a comparative look at the removal of Gaussian plus “salt and pepper” impulse noise from two test images using the proposed method and JSM.

In Figure 4.10 (a) shows the results of Barbara image when it was corrupted by Gaussian noise with sigma is **10** and ratio (r) equals **20%,40%,50%,70%,** or **90%**. First, if the ratio equal **20%,** the JSM was checked as **31.07dB** while the proposed method result is **38.10dB**. Second, if the ratio is equal to **40%**, the JSM was checked as **29.17dB** and the proposed method result is **36.39dB**. Third, if the ratio is equal to **50%**, the JSM was checked as **27.95dB** and the proposed method result is **34.74dB**. Fourth, if the ratio is equal to **70%**, the JSM was checked as **23.99dB** and the proposed method result is **29.63dB**. Finally, if the ratio is equal to **90%**, the JSM was checked as **22.37dB** and the proposed method result is **22.61dB**.

Figure 4.10 (b) shows the results of House image when it was corrupted by Gaussian noise with sigma is **10** and ratio (r) equals **20%,40%,50%,70%,** or **90%**. First, if the ratio is equal to **20%,** the JSM was checked as **34.66dB** and the proposed method result is **37.30dB**. Second, if the ratio is equal to **40%**, the JSM was checked as **34.33dB** and the proposed method result is **36.30dB**. Third, if the ratio is equal to **50%**, the JSM was checked as **33.72dB** and the proposed method result is **35.44dB**. Fourth, if the ratio is equal to **70%**, the JSM was checked as **32.25dB** and the proposed method result is **33.42dB**. Finally, if the ratio is equal to **90%**, the JSM was checked as **27.48dB** and the proposed method result is **27.84dB**.

Summarily, it is evident that the proposed method significantly improves the results of Gaussian plus salt-and-pepper impulse noise when compared to JSM because the PSNR value of the proposed method is considerably higher than in JSM.



(a)

(b)

Figure 4.10: Results of mixed Gaussian plus salt-and-peppers impulse noise removal when compared with JSM for the cover-images (a) Barbara and (b)House.

Chapter 5

CONCLUSION

5.1 Conclusion

Image restoration involves the reduction or outright eradication of the degradation of an image in an effort to enhance the image and recover its original form. The goal of recovering the original image is what distinguishes image restoration from image enhancement, which simply aims to improve the aesthetic appeal of the image and one of the foremost methods for image restoration is JSM.

This thesis proposes a new method for image restoration based on JSM. The proposed method also finds its basis in the effective statistical characterization of the nonlocal self-similarity and local smoothness intrinsic to natural images. In an effort to improve the image restoration results gotten from JSM, the proposed method involves the addition of a SMF to JSM and a MF at the end of every iteration in the restoration process.

Overall, the proposed method makes three major contributions: it establishes JSM in a domain for adjective hybrid space-transformation; using JSM, it develops a new type of minimization functional for solving the image inverse problem within a regularization-based framework and it improves JSM by developing a new rule based in the Split Bregman method.

The proposed method was tested for three applications of image restoration: image deblurring, image inpainting (text removal), and the removal of mixed Gaussian and salt-and-pepper noise. Experimental results indicate that image restoration using the proposed method is significantly improved relative to simple JSM. Furthermore, the convergence of the proposed method was also considerably improved relative to JSM.

5.2 Direction for Future Research

Despite the undoubtedly valuable contribution made by the proposed method to the field of image restoration, there is still room for future research to build on the proposed method in a number of ways.

For example, the proposed method could be enhanced so as to increase the speed of image inpainting without corresponding decreases in the PSNR value of the restored image. Additionally, future research could also include a multi-scale investigation of natural image statistics and even extend the applicability of the proposed method beyond natural images – for example, to video restoration through the incorporation of skin detection methods.

REFERENCES

[1] Banham, M. R., & Katsaggelos, A. K. (1997). Digital image restoration *IEEE Signal Processing Magazine* IEEE (Trans.), *14*(2), 24-41.

[2] Rani, S., & Kumar, D. (2014). A Case Study on Soft Computing Techniques Used for Diabetes Mellitus. *International Journal*, *4*(7).

[3] Mohapatra, B. R., Mishra, A., & Rout, S. K. (2014). A comprehensive review on image restoration techniques. *International Journal of Research in Advent Technology*, *2*(3), 101-105.

[4] Rani, D. D., & Kumar, D. M. (2016). Digital image restoration by usingthe joint statistical model. *IJCSMC*, *5*(7), 225-233.

[5] Rani, S., Jindal, S., & Kaur, B. (2016). A Brief Review on Image Restoration Techniques. *International Journal of Computer Applications*, *150*(12).

[6] Zhang, J., Zhao, D., Xiong, R., Ma, S., & Gao, W. (2014). Image Restoration Using Joint Statistical Modeling in a Space-Transform Domain. *IEEE Trans. Circuits Syst. Video Techn.*, *24*(6), 915-928.

[7] O’Handley, D. A., & Green, W. B. (1972). Recent developments in digital image processing at the image processing laboratory at the Jet Propulsion Laboratory. *Proceedings of the IEEE*, *60*(7), 821-828.

[8] Stockam, T. G., Cannon, T. M., & Ingebresten, R. B. (1975). Blind deconvolution through digital signal processing. *Proceedings of the IEEE*, 63(4), 678-692

[9] (2017) Wikipedia. [online]. https://en.wikipedia.org/wiki/Deblurring.

[10] Sitara, K., & Remya, S. (2012). Image deblurring in bayesian framework using template based blur estimation. *The International Journal of Multimedia & Its Applications*, *4*(1), 137.

[11] Tekalp, A., Kaufman, H., & Woods, J. (1986). Identification of image and blur parameters for the restoration of noncausal blurs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *34*(4), 963-972.

[12] Sonu, J., Akhilesh, D., Diljeet, S., C., Prabhat, K., & S. (2014). Image Deblurring from blurred images. *International Journal of Advanced Research in Computer Science & Technology (IJARCST)*, *2*(3), 41-45.

[13] Yuan, L., Sun, J., Quan, L., & Shum, H. Y. (2007). Image deblurring with blurred/noisy image pairs. In *ACM Transactions on Graphics (TOG)*, 26(3), 1.

[14] Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, *60*(1-4), 259-268.

[15] Ramanaiah, K. V. (2015). Image Restoration Using Joint Statistical ModellingIn Space-Transform Domain. *International Journal of New Technologies in Science and Engineering*, 2(1), 233-241.

[16] Vaseghi, S. V., & Milner, B. P. (1995). Speech recognition in impulsive noise, Inst.of Acoustics, speech and signal processing. In *IEEE Proceedings Integracija Conf*, *Acoustics, speech and signal processing*, *ICASSP-95*, 1, 437-440.

[17] Vaseghi, S. V., & Rayner, P. J.W. (1990). Detection and suppression of impulsive noise in speech communication systems. *IEE Proceedings I Communications, Speech and Vision*, 137(1), 38-46.

[18] Sathua, S. K., Dash, A., & Behera, A. (2017). Removal of Salt and Pepper noise from Gray-Scale and Color Images *International Journal of Computer Science Trends and Technology (IJCST)*, *5*(1), 117-126.

[19] Leavline, E. J., & Singh, D. A. A. G. (2013). Salt and pepper noise detection and removal in gray scale images: an experimental analysis. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, *6*(5), 343-352.

[20] Sawant, H. K., & Deore, M. (2010). A comprehensive review of image enhancement techniques. *International Journal of Computer Technology and Electronics Engineering (IJCTEE)*, *1*(2), 39-44.

[21 Vij, K., & Singh, Y. G. (2011). *Comparative Study of Different Techniques of Image Enhancement for Grayscale and Colour Images* (Doctoral dissertation). *International Journal of VLSI and Signal Processing Applications*, 1(2), 112-117.

[22] Jayaraman, S., Esakkirajan, S., & Veerakima, T. DigitalImage processing. In *a McGrawHillEducationprivatelimited, New Delhi* (3e ed, Tat).

[23] Mundhada, S. O., & Shandilya, V. K. (2012). Spatial and transformation domain techniques for image enhancement. *International Journal of Engineering Science and Innovative Technology (IJESIT)*, *1*(2), 213-16.

[24] Zhao, J. L., Zhao, J. F., Ren, D.X., & Yan, S. (2006). AWavelet based algorithm for edge extraction of images. *International Conference on Machine Learning and Cybernetics,* 3803-3806.

[25] Kother, S. M., Arumuga, S., P., Mohamed, M., & S. (2008). Image De-noising using Discrete Wavelet transform. *IJCSNS International Journal of Computer Science and Network Security*, *8*(1), 213-216.

[26] Ye, Z., Mohamadian, H., & Ye, Y. (2007). Infromation measures for biometric identification via 2D discrete wavelet transform. *Proceedings of the 3rd Annals IEEE conference on Automation Science and Engineering*, 835-840.

[27] Nadenau, M. J., Reichel, J., & Kunt, M. (2003). Wavelet based color image compression: Exploiting the contrast sensitivity function. *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, *12*(1), 58-70.

[28] Schoukens, J., Pintelon, R., & Van Hamme, H. (1992). The interpolated fast Fourier transform: A comparative study. *IEEE Transactions on Instrumentation and Measurement*, *41*(2), 226-232.

[29] Stockham, T. G., Cannon, T. M., &Ingebretsen, R. B. (1975). Blinddeconvlution through digital signal processing. *Proceedings of the IEEE*, 63(4), 678-692.

[30] (2018). *MathWorks. [online]*. https: //www.mathworks.com /help/ vision/ ref/ 2dconvolution.html.

[31] Ayers, G. R., & Dainty, J. C. (1988). Iterative blind deconvolution method and its applications. *Optics Letters*, *13*(7), 547-549.

[32] Levin, A., Weiss, Y., Durand, F., & Freeman, W.T. (2011). Understanding blind deconvolution algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* IEEE (Trans.) *Pattern Anal. Mach. Intell*, *33*(12), 2354-2367.

[33] Krishnan, D., & Fergus, R. (2009). Fast image deconvolution using hyper-Laplacian priors. In *Advances in Neural Information Processing Systems*, 1033-1041.

[34] Varanasi, M. K., & Aazhang, B. (1989). Parametric generalizedGaussian density estimation. *Journal of the Acoustical Society of America*, *86*(4), 1404-1415.

[35] Buades, A., Coll, B., & Morel, J. M. (2005). A non-local algorithm for image denoising. *Internationa Confederación on Computer Vision and Pattern Recognition*, 60-65.

[36] Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, *16*(8), 2080-2095.

[37] Manjón, J. V., Coupé, P., Buades, A., Collins, D. L., & Robles, M. (2012). New methods for MRI denoising based on sparseness and self-similarity. *Medical image analysis*, *16*(1), 18-27.

[38] Ranjith, Ch, K. Praveen,P, R, & Sreenivas, B. (2016). Image Restoration using Statistical Modeling in Space-Transform Domain. *International Journal of Research in Advanced Engineering Technologies*, *5*(5), 113-128.

[39] Zhou, M., Chen, H., Paisley, J., Ren, L., Li, L., Xing, Z., & Carin, L. (2012). Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, *21*(1), 130-144.

[40] Elad, M., Starck, J. L., Querre, P., &Donoho, D.L.(2005). Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, *19*(3), 340-358.

[41] Takeda, H., Farsiu, S., & Milanfar,P. (2007). Kernel regresion for image process and reconstruction. *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society* IEEE (Trans.), *16*(2), 349-366.

[42] Li, Y. R., Shen, L., Dai, D. Q., & Suter, B. W. (2011). Framelet algorithms for de-blurring images corrupted by impulse plus Gaussian noise. *IEEE Transactions on Image Processing*, *20*(7), 1822-1837.

[43] Huang, Y., Ng, M. K., & Wen, Y. (2009). Fast image restoration methods for impulse and Gaussian noise removal. *IEEE Signal Processing Letters*, 16(6), 457-460.

[44] Roth, S., & Black, M. J. (2009). Fields of experts. *International Journal of Computer Vision*, 82(2), 205-229.

[45] Pitas, I., & Venetsanopoulos, A. N. (2013). *Nonlinear digital filters: principles and applications* (Vol. 84). Springer Science & Business Media.

[46] Gnoping Qiu, G. (1996). An improved recursive median filtering scheme for image processing. *IEEE Transactions on Image Processing*, *5*(4), 646-647.

[47] Pushpavalli, R., & Sivaradje, G. (2012). Switching median filter for image enhancement. *International Journal of Scientific & Engineering Research*, *3*(2), 1-5.

[48] Prabhn, R., & Santhosh Baboo, S. (2015). Image Restoration based on enhanced Switching Median Filter with NSSK. *ARPN Journal of Engineering and Applied Sciences*, *10*(16), 6826-6836.

[49] Zhou Wang, Z., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE signal processing Magazine*, 26(1), 98-117.

[50] Goldstein, T., & Osher, S. (2009). The split Bregman algorithm for L1regularize problems. *SIAM Journal on Image Sciences*, 2(2), 323-342.

[51] Cai, J. F., Osher, S., & Shen, Z. W. (2009). Split Bregman methods and frame based image restoration. *Multiscale Modeling and Simulation*, *8*, 5057-5071.

APPENDICES

Appendix A: Split Bergman Method

Split Bergman Based Iterative Algorithm for Image Restoration. As a technique for finding solutions to a host of different L1-regularized optimization problems, the Split-Bregman method [50] is particularly efficient in cases of total-variation regularization and is one of the fastest means of resolving Total Variation denoising, convex image segmentation, image reconstruction from Fourier coefficients, amongst other problems.

So the results in the following image-restoration formulation:

$$argmin_x \frac{1}{2} \|Hu - y\|_2^2 + \tau . \psi_{LSM}(\mathcal{U}) + \lambda \, . \; \psi_{NLSM}(\mathcal{U}) \tag{A.1}$$

Where τ and λ are control parameters. Note that the first term of Eq. (A.1) actually represents the observation constraint and the second and the third represent the image prior local and non- local constraints, respectively. Therefore, it is our belief that better results will be achieved by imposing the above three constraints into the ill-posed image inverse problem. In this section, we apply the algorithmic framework of Split Bregman Iteration to solve Eq. (A.1)

$$min_{uR^n} f(u) + g(Gu),$$

where G R^{n*m}, f: $R^n \longrightarrow$ R, g: $R^m \longrightarrow$ R. The split Bregman Iteration works as follows:

Algorithm 1 Split Bregman Iteration (SBI)

```
1. Set k = 0, choose $\mu > 0, d^{(0)} = 0, u^{(0)} = 0, v^{(0)} = 0$.
2. Repeat
3. $u^{(k+1)} = argmin_u f(u) + \frac{\mu}{2}\left\|Gu - v^{(k)} - d^{(k)}\right\|^2$;
4. $v^{(k+1)} = argmin_u f(u) + \frac{\mu}{2}\left\|Gu - v^{(k)} - d^{(k)}\right\|^2$;
5. $d^{(k+1)} = d^{(k)} - \left(Gu^{(k+1)} - v^{(k+1)}\right)$;
6. $k \leftarrow k + 1$;
7. Repeat until the stopping conditions have been met
```

Let us go back to Eq. (8) and point out how to apply SBI to solve it. First, define

$$f(\boldsymbol{u}) = \tfrac{1}{2}\|\boldsymbol{Hu} - \boldsymbol{y}\|_2^2,$$

$$g(\boldsymbol{v}) = g(\boldsymbol{Gu}) = \tau \cdot \Psi_{LSM}(\boldsymbol{u}) + \lambda \cdot \Psi_{NLSM}(\boldsymbol{u}),$$

where

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{x} \end{bmatrix} = \boldsymbol{Gu},\ \boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^N \text{ and } \boldsymbol{G} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{2N \times N}.$$

Therefore, Eq. (8) is transformed to

$$\operatorname{argmin}_{\boldsymbol{u} \in \mathbb{R}^N, \boldsymbol{v} \in \mathbb{R}^{2N}} f(\boldsymbol{u}) + g(\boldsymbol{v}) \text{ s.t. } \boldsymbol{Gu} = \boldsymbol{v}.$$

Invoking SBI, Line 3 in Algorithm 1 becomes:

$$u^{(k+1)} = argmin_u f(u) + \left\|Gu - v^{(k)} - d^{(k)}\right\|_2^2 = \frac{1}{2}\|Hu - y\|_2^2 + \frac{\mu}{2}\left\|\begin{bmatrix} I \\ I \end{bmatrix} u - \begin{bmatrix} w^{(k)} \\ x^{(k)} \end{bmatrix} - \begin{bmatrix} b^{(k)} \\ c^{(k)} \end{bmatrix}\right\|_2^2 \tag{A.2}$$

Where $d^{(k)} = \begin{bmatrix} b^{(k)} \\ c^{(k)} \end{bmatrix} \in \mathcal{R}^{2N}, b^{(k)}, c^{(k)}, \in \mathcal{R}^N$ Splitting *l2* norm in Eq. (A.2), we have

$$u^{(k+1)} = argmin_u \frac{1}{2}\|Hu - y\|_2^2 - y \ \Big\|_2^2 + \frac{\mu}{2}\Big\| \ u - \ w^{(k)} - \ b^{(k)} \ \Big\|_2^2 + \frac{\mu}{2}\Big\| \ u + \frac{\mu}{2}\|u - \ x^{(k)} - \ c^{(k)}\|_2^2, \tag{A.3}$$

Next, Line 4 in Algorithm 1 becomes:

$$u^{(k+1)} = \begin{bmatrix} w^{(k+1)} \\ x^{(k+1)} \end{bmatrix} = argmin_{w,x} \Big\{ \tau.\psi_{LSM}(w) + \ \lambda \ . \ \psi_{NLSM}(x) + \frac{\mu}{2}\ \|u^{(k+1)} - w - \ b^{(k)}\|_2^2 + \ \frac{\mu}{2}\ \|u^{(k+1)} - w\ c^{(k)}\|_2^2 \Big\} \tag{A.4}$$

Clearly, the minimization with respect to w x, are decoupled, thus can be solved separately, leading to

$$w^{(\boldsymbol{k+1})} = \ argmin_w \ \tau.\psi_{LSM}(w) + \ \frac{\mu}{2}\ \|u^{(k+1)} - w - \ b^{(k)}\|_2^2 \tag{A.5}$$

$$x^{(\boldsymbol{k+1})} = \ argmin_x \ \lambda\, .\psi_{NLSM}(w) + \ \frac{\mu}{2}\ \|u^{(k+1)} - x - \ c^{(k)}\|_2^2 \tag{A.6}$$

According to line 5 in algorithms 1, the update of d_k

$$d^{(k+1)} = \begin{bmatrix} b^{(k+1)} \\ c^{(k+1)} \end{bmatrix} = \begin{bmatrix} b^{(k)} \\ c^{(k)} \end{bmatrix} - (\begin{bmatrix} I \\ I \end{bmatrix} u^{(k+1)} - \begin{bmatrix} w^{(k+1)} \\ x^{(k+1)} \end{bmatrix}), \tag{A.7}$$

This can be simplified into the following two expressions:

$$b^{(k+1)} = \ b^{(k)} - (u^{(k+1)} - w^{(k+1)}),$$

$$c^{(k+1)} = \ c^{(k)} - (u^{(k+1)} - x^{(k+1)}),$$

In summary, the minimization for Eq. (A.1) is useful in the resolution of three sub-problems – u , w and x – based on the Split Bregman Iteration (SBI) [51]. The full algorithm used in solving Eq. (A.1) is outlined in **Algorithm 2**.

Taking into account SBI convergence, the following theorem is used in proving the convergence of the proposed algorithm using JSM in **Algorithm 2**.

Theorem 1. Outlined in **Algorithm 2**, the JSM solves Formulation (A.1) given above.

Proof: The algorithm is evidently a form of the Split Bregman Iteration. Because the three functions f(.), $\psi_{LSM}(.)$ and $\psi_{NLSM}(.)$ are all proper, closed, and convex, the following formulation guarantees the convergence of the proposed algorithm:

Algorithm 2 a complete description of JSM

```
Input: $y$ (observed image) and $H$ (linear matrix operator)
Initialization: k=0, $u^{(0)} = y, b^{(0)} = c^{(0)} = w^{(0)} = 0, \mu, \tau, \lambda$
Repeat;
$u^{(k-1)} = argmin_u \frac{1}{2}\|Hu - y\|_2^2 + \frac{\mu}{2}\left\|u - w^{(k)} - b^{(k)}\right\|_2^2 + \frac{\mu}{2}\left\|u - x^{(k)} - c^{(k)}\right\|_2^2$
$p^{(k)} = u^{(k+1)} - b^{(k)}; y = {}^{\tau}/_{\mu};$
$w^{(k+1)} = prox_{\gamma}(\psi_{LSM})(p^{(k)});$
$r^{(k)} = u^{(k+1)} - c^{(k)}; \alpha = {}^{\lambda}/_{x}$
$x^{(k+1)} = proX_{a}(\psi_{NLSM})(r^{(k)});$
$b^{(k+1)} = b^{(k)} - (u^{(k+1)} - w^{(k+1)});$
$c^{(k+1)} = c^{(k)} - (u^{(k+1)} - x^{(k+1)});$
Until the stopping conditions have been met.
Output: $\boldsymbol{u}$ (the restored final image).
```

Appendix B: Joint Statistical Modeling (JSM)

Three major contributions are made here. First, it establishes, in a domain for adjective hybrid space-transformation, joint applied mathematics modeling JSM; this offers a way through which nonlocal self-similarity could be mixed with native smoothness such that the result simultaneously corroborates numerous reliable and equally robust estimations. Second, a new type of minimization functional to be used in solving the image inverse problem is developed within a regularization-based framework using JSM. Finally, a new rule based in the Split Bregman method is also developed in an effort to create a strong and tractable JSM; the rule is intended to resolve several prospective drawbacks that relate to theory-based evidence of convergence [6].

B.1 We have three sub-problems:

B.1.1 *u* sub-problem

In an effort to increase the flexibility of the solution for Eq. (A.3), two parameters $-$ $\mu_1 and\ \mu_2$ $-$ are introduced to replace, as μ does not include the convergence of the algorithm. So, given *w,x*, the sub-problem u outlined in Eq. (A.3) is rewritten as follows:

$$u = \ argmin_u \frac{1}{2}\|Hu - y\|_2^2 + \frac{\mu 1}{2}||u - \ w^{(k)} - \ b^{(k)} \Big\|_2^2 + \frac{\mu 2}{2}\Big\| u \ - \ x^{(k)} \\ - \ c^{(k)}||_2^2 \qquad \text{(B.1)}$$

B.1.2 *w* sub-problem

w sub-problem, the proximal map associated to $\psi_{LSM}(.)$, can be regarded as a denoising filtering with anisotropic total variation as mentioned before . To solve it, one of the intrinsic overcome this difficulty, Chambolle suggested to consider adual approach, and developed a globally convergent gradi-Laplacian, or GGD process, which is more general. In our experiments, we exploit a fixed number of iterations of FISTA to solve w sub-problem, which is computationally efficient and empirically found not to compromise convergence of the proposed algorithm.

B.1.3 x sub-problem

Taking into account *w, u*, the sub-problem x s expressed as[43]:

$$x = prox_a\ (\psi_{NLSM})(x) = \ argmin_x \left\{\frac{1}{2}\ \|x - r\|_2^2 + a.\psi_{NLSM}(x)\ \right\}$$

$$argmin_x \left\{\frac{1}{2}\ \|x - r\|_2^2 + a.\|\Theta_x\|_1\right\} \tag{B.2}$$

Consequently, the closed solution form of the sub-problem x in Eq. (B.2) is written as:

$$x{=}\Omega_{NLSM}(\Theta_x) = \Omega_{NLSM}\left(soft\left(\Theta_r, \sqrt{2p}\right)\right) \tag{B.3}$$

The consequence of taking r to be a noisy form of x is that the statistics of e= $x - r$ then need to be experimentally investigated.