Improved Algorithms for Image Super-Resolution

by

Diego Ariel Sorrentino

B.Eng (Hons), Buenos Aires Institute of Technology, 2004

A Thesis Submitted in Partial Fullfillment of the
Requirements for the Degree of

Master of Applied Science

in the Department of Electrical and Computer Engineering

Improved Algorithms for Image Super-Resolution

by

Diego Ariel Sorrentino

B.Eng (Hons), Buenos Aires Institute of Technology, 2004

Supervisory Committee

Dr. Andreas Antoniou, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Pan Agathoklis, Department Member
(Department of Electrical and Computer Engineering)

Dr. Colin Bradley, Outside Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. Andreas Antoniou, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Pan Agathoklis, Department Member
(Department of Electrical and Computer Engineering)

Dr. Colin Bradley, Outside Member
(Department of Mechanical Engineering)

ABSTRACT

The image super-resolution (SR) problem is a generalization of the image restoration problem which is concerned with blur, noise, and aliasing effects. In the context of digital imaging, the purpose of image SR algorithms is to compensate for degradations such as blur resulting from camera motion and inaccurate focusing, sensor noise, and undersampling.

Multiframe image SR algorithms can be used to obtain a higher-quality higher-resolution (HR) image by fusing several images that are sub-pixel-shifted versions of the same scene. By means of these algorithms, the task of super-resolving an image is often approached as an inversion problem in which a set of low-quality low-resolution (LR) images is considered to be the result of processing a high-quality high-resolution image through a dynamic image acquisition model. A special class of SR algorithms, known as *'maximum-likelihood super-resolution'* (MLSR) algorithms, utilize a stochastic approach for the inversion of such a model. Basically, the HR image that is most likely to produce the observed LR images is found by solving an optimization problem.

In this thesis, an overview of the most representative SR algorithms is presented. Then, the performance of two state-of-the-art MLSR algorithms based on steepest-descent optimization for grayscale and color images is evaluated and later improved by the introduction of sophisticated quasi-Newton optimization algorithms.

The Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithms are appropriately reformulated to cope with the large-scale nature of the SR problem and are then applied to the SR schemes. Experimental results show that by means of the proposed algorithms, grayscale reconstruction is considerably accelerated and the quality achieved in color SR is significantly improved. Moreover, by the introduction of a practical inexact line search, the need for selecting an important parameter is eliminated.

Storage-efficient variants of the BFGS algorithm are also investigated. SR algorithms based on the memoryless BFGS (MBFGS) and limited-memory BFGS (LBFGS) methods are formulated. Experimental results indicate that the proposed algorithms, like the BFGS algorithm, perform the grayscale reconstruction considerably faster and obtain color images of better quality. At the same time, the storage requirements for the MBFGS are comparable to those of the steepest-descent based algorithms while the LBFGS algorithm offers a meaningful trade-off between reconstructed image quality and storage requirements.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AWGN	additive white Gaussian noise
BFGS	Broyden Fletcher Goldfarb Shanno
CCD	charge-coupled device
CFA	color filter array
CFT	continuous Fourier transform
CPU	central processing unit
DFP	Davidon Fletcher Powell
DFT	discrete Fourier transform
DPSNR	difference in peak signal-to-noise ratio
DWT	discrete wavelet transform
fpm	floating point multiplications
HDTV	high-definition television
HR	high-resolution
ILS	inexact line search
LBFGS	limited memory BFGS
LR	low-resolution
MAP	maximum-a-posteriori
MBFGS	memoryless BFGS
ML	maximum likelihood
MLSR	maximum likelihood super-resolution
MMSE	minimum mean-squared error
PSF	point spread function
PSNR	peak signal-to-noise ratio
RTLS	recursive total least-squares
SD	steepest-descent

SDTV	standard-definition television
SNR	signal-to-noise ratio
SR	super-resolution
TLS	total least-squares

Acknowledgment

First and foremost, I would like to express my gratitude to my supervisor, Dr. Andreas Antoniou, for supporting my research throughout these years. It has been an enormous privilege to work under the guidance of one of the greatest contributors to the digital signal processing field. His emphasis on independent research and good writing style helped me become a better professional.

During the course of my studies, personal circumstances required me to put research aside for some time. Dr. Antoniou was very understanding and supportive on this matter and for that my family and I will be always immensely grateful.

Research work has many ups and downs, and support is fundamental to keep oneself focused. I would not have gotten far without the love, patience, and understanding of my wife, Laura, to whom I dedicate this work. Her encouragement was the most influential driver of this achievement.

Many thanks go to my dearest friends in Victoria—my family when far from homeland—Julio, Jany, Caroll, Valeria, Carmen, Antonio, Carlos, and Karla. Thanks for all those memorable moments.

I would also like to thank my officemates and labmates: Paramesh, Haytham, Solmaz, and Marsonno. Special thanks go to Najith for sharing his philosophy of life with me and for his valuable support in stressful times.

Thanks to Dr. Pan Agathoklis and Dr. Wu-Sheng Lu for being always kindly available for consultations throughout the completion of this work.

I appreciate the help received from the office and technical staff of the Department of Electrical and Computer Engineering.

My gratitude also goes to professors Roxana Saint-Nom and Daniel Jacoby at the Buenos Aires Institute of Technology for introducing me to the fascinating world of digital signal processing.

Finally, as in every step of my life, I will be greatly indebted to the support and sacrifice of my parents. This is for you too.

Diego A. Sorrentino, Victoria, BC, Canada

Dedication

To Laura, love of my life.

Chapter 1

Introduction to Image Super-Resolution

1.1 Background

As the number of electronic-imaging-based applications grows so does the need for the acquisition of higher resolution images. Increasing the acquisition resolution of optical systems improves their capability to capture fine details pertaining to the scene being observed. Applications like medical imaging, surveillance, astronomical imaging and remote sensing can greatly benefit from the availability of highly detailed images.

Since the 1970's, charged-coupled devices (CCDs) and CMOS sensors have been widely used for digital image acquisition. Even though these sensors are suitable for many applications, there is an increasing demand for high-resolution imaging systems both in the amateur and scientific markets [1, 2]. A straightforward procedure to improve the resolution capabilities of an image acquisition system is to increase the pixel count of its sensor array. This can be achieved either by making each sensing element smaller or by increasing the size of the sensor array. Unfortunately, both these procedures are limited by sensor technology. As the sensing element size decreases so does the amount of light received while the inherent noise level of the sensor remains

constant. Consequently, reducing the pixel size results in a degradation of the signal-to-noise ratio (SNR) of the acquired image. On the other hand, if the sensor array size is increased while maintaining the sensing element size, the array capacitance increases thereby slowing down the charge transfer rate [1, 3].

A promising alternative to increase the spatial resolution capability of an imaging system while using inexpensive low-resolution (LR) sensors is provided by digital image processing techniques known as 'super-resolution' (SR). These techniques produce a high-quality, high-resolution (HR) image by appropriately extracting HR information from several LR images of a scene.

In what follows we introduce some fundamental concepts related to image super-resolution and its applications. Next, an outline of the contents and contributions of this thesis is given.

1.2 Fundamental Concepts

1.2.1 Low-Resolution Input Images

Image super-resolution algorithms produce one high-quality, high-resolution image from a set of low-quality, low-resolution images. This is a process in which spatial resolution is gained at the expense of needing several distinct views of the same scene [4]. As illustrated in Fig. 1.1, there are many practical situations in which it is possible to obtain different views of the same scene. These can be

- images pertaining to the same scene acquired with the same camera from different viewpoints at different times,
- images pertaining to the same scene acquired at the same time but with different cameras placed in different viewpoints,

- image frames extracted from a video sequence.

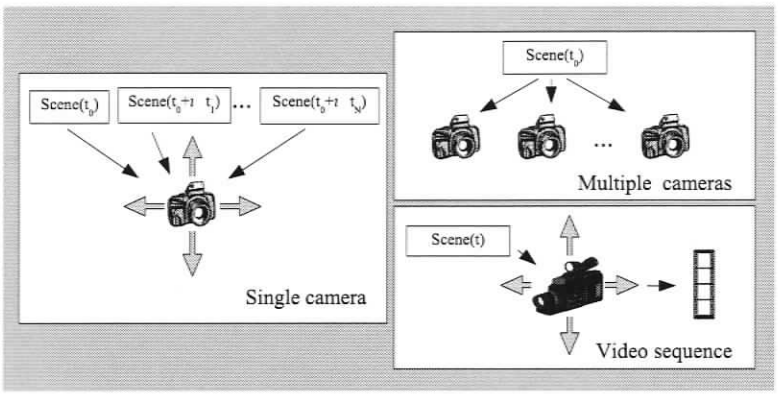


Figure 1.1: Ways to obtain the LR image set

Except for some wavelet-based methods, a fundamental requirement for increasing the spatial resolution is that each LR image available should provide a different view of the scene. In other words, each LR image should provide some information which is not available in the rest of the set. This requirement is usually translated into the need for sub-pixel displacement between the acquired LR images.

1.2.2 Image-Acquisition Model

To successfully recover a high-quality HR image from its corresponding LR images it is important to understand and accurately model how these LR images are obtained. In fact, image super-resolution can be regarded as an inverse problem in which we attempt to reverse the effects of a series of degradations.

Figure 1.2 illustrates an image acquisition model that accounts for the processes that take place when acquiring a series of digital images. The observed scene is first represented by a 2-dimensional light intensity analog signal. Subsequently, a sampled version of the analog signal is obtained using a sufficiently high sampling frequency to avoid aliasing effects. This artificially introduced sampling process serves as a

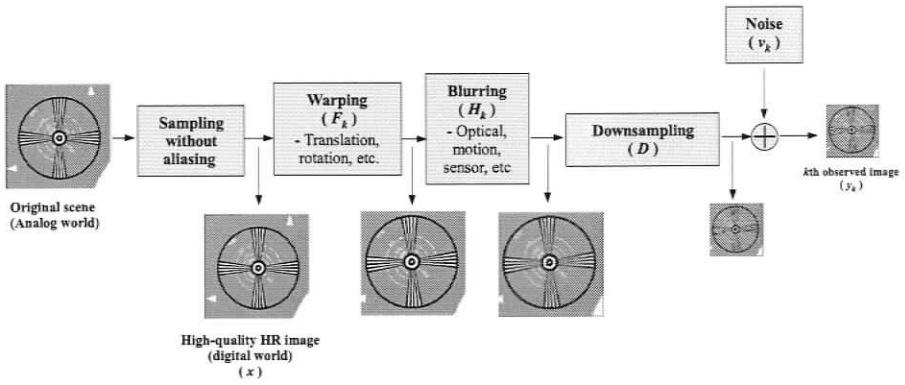


Figure 1.2: Image acquisition model.

bridge between the analog and the digital worlds. Next, the effect of the camera motion (translation, rotation, tilt, etc.) that takes place in between each acquisition is modelled. After that, a blurring effect resulting from the combined influence of lens thickness, nonzero aperture time, and nonnegligible sensor size is introduced. A downsampling process produces an image with the same number of pixels as the number of pixels available at the light sensor array. The model also accounts for inherent additive sensor noise. In summary, the image acquisition model takes us from an analog respresentation of a scene to a series of warped, blurry, aliased, and noisy observations.

Lexicographical ordering is often used in image processing, particularly in image super-resolution [5]. By virtue of this notation, an image is represented by a column vector which is an ordered raster scan of the image rows. In this way, an $M \times L$-pixel image is represented by an ML-element column vector as shown in Fig. 1.3. Using lexicographically ordered notation, the process by which a set of N low-quality, low-resolution images, each comprising $M \times L$ pixels, is acquired by an imaging system can be modeled mathematically as

$$\boldsymbol{y_k} = \boldsymbol{D}\boldsymbol{H_k}\boldsymbol{F_k}\boldsymbol{x} + \boldsymbol{v_k} \quad k = 1, \ldots, N \tag{1.1}$$

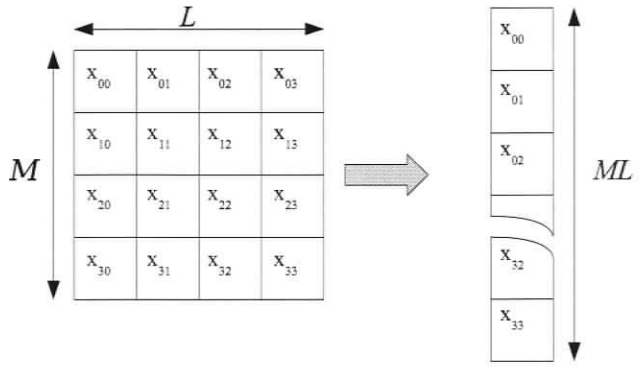


Figure 1.3: Lexicographically ordered notation.

where $\boldsymbol{x}$ is an r^2ML-element column vector representing a raster scan of the HR image, $\boldsymbol{y_k}$ and $\boldsymbol{v_k}$ are ML-element column vectors representing the observed frames and additive noise sources, respectively, $\boldsymbol{D}$ is an $ML \times r^2ML$ matrix that represents the downsampling operation, $\boldsymbol{H_k}$ is an $r^2ML \times r^2ML$ matrix that models the point-spread function of the camera, $\boldsymbol{F_k}$ is an $r^2ML \times r^2ML$ matrix that models the camera motion corresponding to each acquired frame, and r is known as the resolution enhancement factor.

1.3 Existing Super-Resolution Techniques

Since the seminal work of Tsai and Huang [6], a significant number of methods have been proposed by the research community [2, 7]. Existing methods range from those based on the spatial domain to those based on the frequency and wavelet domains, from the ones that are purely deterministic to those based on stochastic models. Some methods can invert the observation model without any information about the nature of camera movement and blurring whereas others may require information about the underlying processes involved in the image acquisition. An overview of the most representative super-resolution methods is provided in Chapter 2.

Super-resolution techniques find applications in many fields. In surveillance, for example, synthetic zooming is made possible by the reconstruction of HR images from several LR images. In computer vision applications, like license plate readers and face recognition systems, accuracy can be improved by using additional information that is recovered from several LR images. Remote sensing applications are specially suited for image super-resolution. In these applications it is common to find image sequences pertaining to the same scene and, therefore, it is easy to apply super-resolution algorithms to increase the resolution power. Sub-pixel displacement information, available in compression schemes that describe a video sequence in terms of global and local motion vectors, can be exploited to achieve a higher quality reconstruction (decompression) [8]. Yet another emerging application of super-resolution algorithms, driven by the need to display standard-definition TV (SDTV) images in high-definition receivers (HDTV) without visual artifacts, is the conversion of SDTV signals to HDTV signals.

1.4 Contributions and Outline of the Thesis

In this section, we summarize the contributions of this thesis.

Chapter 2 provides a review of some of the most representative super-resolution techniques proposed to date. In this chapter, we divide these techniques into three categories: those based on frequency-domain formulation, those based on spatial-domain formulation, and those based on wavelet domain formulation. We describe these techniques and compare them in terms of their algorithmic efficiency, their ability to incorporate previous information about the desired solution as well as their ability to account for realistic image observation models.

In Chapter 3, we study two state-of-the-art spatial-domain based algorithms reported in [9] and [10]. These algorithms are intended for grayscale and color super-

resolution, respectively, and are based on the steepest-descent minimization of multi-term objective functions. In this chapter, we give details on their implementation and the fundamentals behind their operation. We then show the results of simulations aimed at assessing their effectiveness and argue that these algorithms rely on a simple minimization scheme which is innappropriate given the ill-conditioned nature of the super-resolution problem. Additionally, we observe that a key parameter has to be manually tuned to achieve good results and, consequently, the overall performance of these algorithms is dependant on its proper selection.

In Chapter 4, we address the deficiencies of the algorithms studied in Chapter 3. Specfically, we design alternative algorithms based on quasi-Newton optimization methods which are known to have excellent convergence properties. To cope with the inherent large-scale nature of the super-resolution problem, we propose a reformulation of two standard algorithms, namely, the Davidon-Fletcher-Powell (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS). These reformulations are based on convenient outer product expansions and can be used to calculate search directions without having to deal with large-scale matrix-vector products. The need for manual adjustment of a key parameter is eliminated in both algorithms by the introduction of a practical inexact line search that adaptively selects this parameter. Towards the end of Chapter 4, we present the results of comparative experiments aimed at assessing the benefits of adopting the proposed optimization schemes. These results indicate that an improvement in the effectiveness of these algorithms over algorithms based on the steepest-descent (SD) method is achieved for both grayscale and color super-resolution. In the first case, a noticeable improvement in convergence time is observed at the expense of additional storage requirements. On the other hand, results for color super-resolution indicate that a significant improvement in the quality of the reconstructed images, as measured by the peak signal-to-noise ratio (PSNR), can be achieved at the expense of additional storage requirements and,

possibly, additional computation time depending on the image size.

In Chapter 5, we present two algorithms based on the memoryless BFGS (MBFGS) and the limited-memory BFGS (LBFGS) methods [11] [12]. These methods inherit many of the desirable convergence properties of the standard BFGS method while requiring less storage and computational capabilities. Results of the application of the proposed algorithms to sample images indicate that for grayscale super-resolution the MBFGS-based algorithm provides the best overall improvement in terms of reconstructed image quality, storage requirements, and convergence speed when compared to an SD approach and other quasi-Newton-based algorithms. On the other hand, for color super-resolution, experimental results show that the LBFGS-based algorithm constitutes an effective alternative that leads to a meaningful compromise between the amount of storage and computational effort required, and the quality of the images obtained.

Finally, in Chapter 6, we provide concluding remarks as well as suggestions for future research.

Chapter 2

Overview of Existing Image Super-Resolution Algorithms

2.1 Introduction

In this chapter, we review some of the existing techniques for image super-resolution. The purpose of this chapter is not to provide an exhaustive review but rather to give an idea of the different angles from which the image super-resolution problem has been tackled. We will follow to a great extent the taxonomy proposed in [7] with the addition of some wavelet-domain methods. Consequently, we will divide image super-resolution algorithms into three main categories: those based on the frequency domain, those based on the spatial domain, and those based on the wavelet domain.

2.2 Frequency-Domain Methods

2.2.1 Alias-Removal Reconstruction

One of the earliest attempts to reconstruct a high-resolution (HR) image from several low-resolution (LR) images was in a 1984 paper by Tsai and Huang [6]. Their work

was motivated by the need to reconstruct an HR image from several LR images acquired from a LANDSAT satellite. In [6], Tsai and Huang assumed translational motion between the satellite images and ignored further degradations such as blur or noise. Then, by exploiting the shift property of the continuous Fourier transform (CFT) and the relationship between the discrete Fourier transform (DFT) and the CFT, they proposed a procedure for recovering samples of the CFT of the desired HR image from the DFT coefficients of the observed LR images. Then, they used the estimated samples of the CFT to recover the desired HR image by using the inverse DFT. This technique has been referred to as *alias removal reconstruction* and its details are as follows.

Considering the original scene, $f(x, y)$, and its translated versions, $f_k(x, y) = f(x + \delta_{x,k}, y + \delta_{y,k})$, we note that the corresponding CFTs are related by the shift property of the Fourier transform as

$$F_k(u, v) = e^{j2\pi(\delta_{x,k}u + \delta_{y,k}v)} F(u, v) \tag{2.1}$$

where $(\delta_{x,k}, \delta_{y,k})$ is the relative displacement of the kth LR image. From the aliasing relationship between the DFT and CFT of the underlying continuous signal [13] and (2.1), the DFT of the kth LR image at the (m, n) discrete frequency can be expressed as

$$\begin{aligned} F_{mn}(k) &= \frac{1}{T_x T_y} \sum_{j=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} F_k\left(\frac{m}{MT_x} + j\omega_{sx}, \frac{n}{NT_y} + l\omega_{sy}\right) \\ &= \frac{1}{T_x T_y} \sum_{j=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} e^{j2\pi(\delta_{x,k}u + \delta_{y,k}v)} F\left(\frac{m}{MT_x} + j\omega_{sx}, \frac{n}{NT_y} + l\omega_{sy}\right) \end{aligned} \tag{2.2}$$

Then, assuming that $f(x, y)$ is bandlimited and that the size of each LR frame is $M \times N$ pixels, (2.2) can be rewritten in terms of MN different linear systems of equations of the form

$$\boldsymbol{g}_{mn} = \boldsymbol{\Phi}_{mn} \tilde{\boldsymbol{f}}^{mn} \tag{2.3}$$

where the kth component of column vector $\boldsymbol{g_{mn}}$ is $\boldsymbol{g_{mn}}(k) = F_{mn}(k)$, $\boldsymbol{\Phi_{mn}}$ contains information regarding the relative motion of the acquired frames, and column vector $\tilde{\boldsymbol{f}}^{\boldsymbol{mn}}$ contains regularly spaced samples of $F(u, v)$, the CFT of the HR image.

Next, Tsai and Huang proposed an efficient technique to solve the linear systems of equations by finding a convenient factorization for $\boldsymbol{\Phi_{mn}}$. In this way, the super-resolution problem solution reduces to calculating the DFT of the observed images, estimating the CFT of the HR image by solving NM different systems of equations like that in (2.3), and recovering the HR image by means of the inverse DFT.

Although the approach of Tsai and Huang is computationally very attractive, the image acquisition model is very simple. Noise and blur are not considered and the global translation model is not appropriate for many applications. Some of these limitations are addressed in works like [14] and [15].

2.2.2 Recursive Least-Squares Techniques

In so-called recursive least-squares techniques, the super-resolution problem based on (2.3) is approached in an iterative manner to improve computational efficiency. For example, in [16] the Tsai and Huang approach is extended to account for blur and noise. In this work a least-squares solution for (2.3) is found through the unconstrained minimization

$$\underset{\tilde{\boldsymbol{f}}^{\boldsymbol{mn}}}{\text{Minimize}} \, \|\boldsymbol{\Phi_{mn}}\tilde{\boldsymbol{f}}^{\boldsymbol{mn}} - \boldsymbol{g_{mn}}\|_2^2 \tag{2.4}$$

which can be tackled using iterative methods.

Zeros in the frequency response of the blurring function not only prevent us from recovering certain frequencies but also severely degrade the numerical conditioning of the problem in (2.4). To account for these shortcomings, a regularization functional,

$\gamma(\cdot)$, is often introduced by reformulating (2.4) as [17]

$$\underset{\tilde{\boldsymbol{f}}^{mn}}{\text{Minimize}} \|\boldsymbol{\Phi}_{mn}\tilde{\boldsymbol{f}}^{mn} - \boldsymbol{g}_{mn}\|_2^2 + \gamma(\tilde{\boldsymbol{f}}^{mn}) \tag{2.5}$$

2.2.3 Recursive Total Least-Squares Methods

Recursive total least-squares (TLSs) methods [18] are extensions of the least-squares techniques discussed earlier. The main contribution of these methods is that they incorporate errors resulting from the motion estimation process to the image acquisition model. In this way, the LR discrete frequencies can be written as [19]

$$\boldsymbol{g}_{mn} = (\boldsymbol{\Phi}_{mn} + \boldsymbol{E}_{mn})\,\tilde{\boldsymbol{f}}^{mn} + \boldsymbol{v}_{mn} \tag{2.6}$$

where $\boldsymbol{E}_{mn}$ is a perturbation matrix modelling errors in the motion estimation, and $\boldsymbol{v}_{nm}$ models observation noise. The TLS problem may be formulated as the constrained minimization problem

$$\begin{aligned} &\text{Minimize} \quad \|\boldsymbol{v}_{mn}\|_F + \|\boldsymbol{E}_{mn}\|_F \\ &\text{subject to:} \quad \boldsymbol{g}_{mn} - \boldsymbol{v}_{mn} = (\boldsymbol{\Phi}_{mn} + \boldsymbol{E}_{mn})\,\tilde{\boldsymbol{f}}^{mn} \end{aligned} \tag{2.7}$$

A recursive total least-squares (RTLS) algorithm based on the adaptive filtering work described in [20] can be used to solve the problem in (2.7) [19]. Even though the approach is suitable for massive parallel computing and can incorporate space invariant blur, it cannot incorporate scene or camera motion models beyond simple translations.

2.2.4 Multichannel-Sampling-Theorem-Based Techniques

These algorithms are based on the Brown multichannel-sampling theorem [21] which is a simpler formulation of the Papoulis generalized sampling theorem [22].

Basically, the Papoulis generalized sampling theorem states that a bandlimited signal $f(x)$ for which $F(\omega) = 0$ for $|\omega| > \sigma$ can be expressed in terms of the samples $g_k(nT)$, $k = 1, \ldots, m$, $\infty \leq n \leq \infty$ that result from sampling at the sub-Nyquist rate of $T = m(2\pi/\sigma)$ the output of m independent linear systems to which $f(x)$ is applied. Papoulis [22] and later Brown [21] have shown how to find the m reconstruction filters Y_k to obtain $f(x)$ from the samples $g_k(nT)$ so that

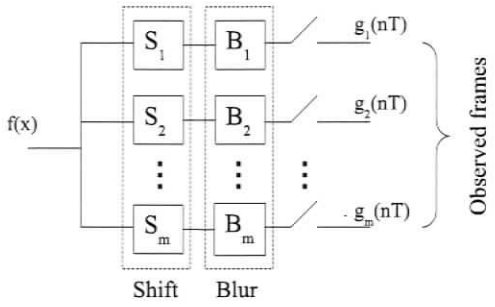
$$f(x) = \sum_{k=1}^{m} \sum_{n=-\infty}^{\infty} g_k(nT) y_k(x - nT) \tag{2.8}$$

Ur and Gross [23] were the first to put the super-resolution reconstruction problem in the context of the multichannel-sampling theorem. As shown in Fig. 2.1(a), the acquisition of the LR frames can be modeled as a multichannel operation. Then, if constant and space-invariant blur is assumed the process can be simplified and represented as in Fig. 2.1(b).

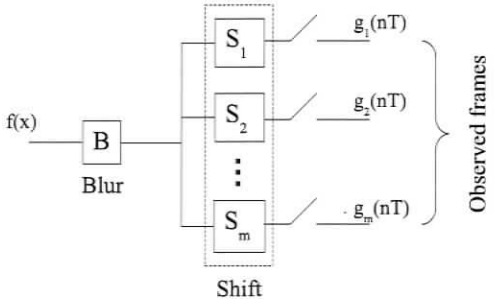
In [23] the reconstruction of the HR image is performed in two steps. First, the LR frames are merged by the reconstruction filters Y_k. Then, the merged image is deblurred using a standard deblurring process. The entire process is illustrated in Fig. 2.2.

The algorithm of Ur and Gross is attractive because it is well posed and robust. On the other hand, noise degradation is not considered, it assumes global translational motion, and that all LR frames are affected by the same space-invariant blur. Consequently, the approach may not be suitable for several applications that arise in practise.

A much more powerful algorithm based on multichannel sampling is proposed in [24]. In this paper, the Papoulis sampling theorem is generalized in a multidimensional multiresolution setup framework so that the channel sampling densities no longer have to be equal. In addition, to overcome sampling ill-posedness in the presence of noise,



(a)



(b)

Figure 2.1: (a) Multichannel representation of the image acquisition process. (b) Multichannel representation after space and time invariant blur assumption.

image frames are projected from standard orthonormal bases onto optimal Riesz bases defined by the channel-point spread function (PSF).

To sum up, frequency-domain methods for super-resolution are of simple formulation and are computationally efficient. On the other hand, they fail to incorporate complex motion models and space-dependent degradations. Additionally, it is difficult to incorporate previous knowledge about the solution in the form of spatially dependent regularizing functionals that would improve the conditioning of

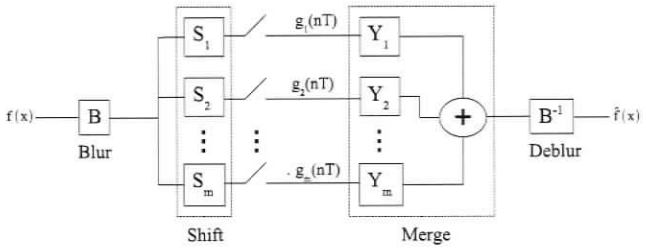


Figure 2.2: Multichannel representation of the image reconstruction process.

the problem. As shown in the next section, these difficulties are successfully addressed by spatial-domain super-resolution methods.

2.3 Spatial-Domain Methods

Spatial-domain super-resolution methods address many of the practical shortcomings of frequency-domain-based algorithms. Two of the more important features of spatial-domain methods is that they can model more complex camera motions and can incorporate previous information about the desired solution in a simple manner. These advantages, however, come at the expense of more complex algorithms that involve a heavier computational load. However, given their advantages spatial-domain methods are usually preferred over frequency-domain methods.

2.3.1 Non-Uniform Interpolation Methods

If time- and space-invariant blur can be assumed, then an HR image can be recovered from several LR images by means of non-uniform interpolation methods. When blur is time-invariant its corresponding operator is constant, i.e., $\boldsymbol{B_k} = \boldsymbol{B}$. In addition, if blur is space-invariant, then the blur operator can conmute with the warping operator, i.e., $\boldsymbol{BF_k} = \boldsymbol{F_kB}$. As a result, the image acquisition process can be mathematically

expressed as

$$\boldsymbol{y_k} = \boldsymbol{DF_kBx} + \boldsymbol{\nu_k} \tag{2.9}$$

$$\boldsymbol{y_k} = \boldsymbol{DF_k\tilde{z}} \tag{2.10}$$

where $\tilde{\boldsymbol{z}}$ is a blurry and noisy estimate of the HR resolution image. Image $\tilde{\boldsymbol{z}}$ can be estimated from the LR images, $\boldsymbol{y_k}$, by appropriately reversing the combined warping and downsampling processes. Finally, the HR image, $\boldsymbol{x}$, can be estimated by debluring and denoising $\tilde{\boldsymbol{z}}$.

Figure 2.3 illustrates the upsampling, motion compensation and interpolation processes. The figure shows three LR frames which after being upsampled and motion compensated are superimposed on an HR grid thus forming a non-uniform sampling composite. Interpolation is then applied to estimate the values of $\tilde{\boldsymbol{z}}$ on the HR grid. Then, the estimated $\tilde{\boldsymbol{z}}$ is deblurred and denoised to recover the original HR image. The entire non-uniform interpolation-based image super-resolution scheme is illustrated in Fig. 2.4.

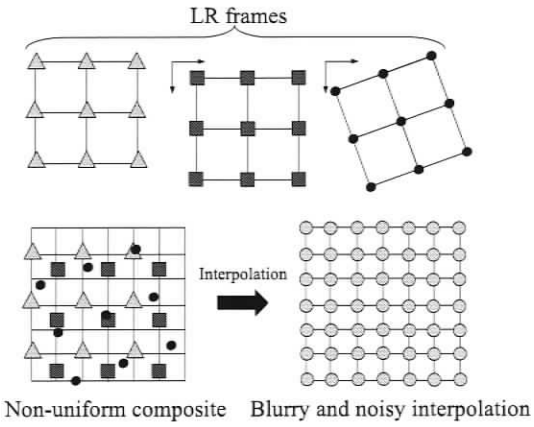


Figure 2.3: Upsampling, motion compensation, and non-uniform interpolation processes.

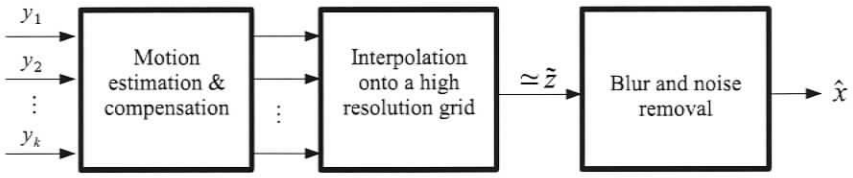


Figure 2.4: Non-uniform-interpolation-based super-resolution scheme.

One of the earliest attempts to reconstruct an HR image using the procedure described is the one proposed by Keren et al. [25]. In this paper, a simple procedure to achieve sub-pixel accurate registration and a way to fuse the registered images into a higher-resolution frame are proposed. Then, a procedure is introduced by which a higher quality estimation of the HR image is obtained by means of numerical optimization.

Another example of the use of non-uniform interpolation is found in the work of Aizawa et al. [26]. In this work, the possibility of increasing the spatial resolution of images obtained with stereo cameras is investigated. Specifically, it is shown how the resolving power of a system consisting of two cameras that provide different views of the same scene goes beyond the individual resolving power of each camera. Given the nature of the stereo pair, the available samples are arranged in a non-uniform grid that results from the regular samples of the reference image among which samples of the second image are scattered. This problem is addressed using a method for reconstruction from non-uniformly spaced samples developed in [27].

In [28], Nguyen and Milanfar introduce a wavelet-based interpolation-restoration method for super-resolution. When the camera motion is purely translational, or has been corrected to be so, the resulting type of non-uniform sampling is known as interlaced sampling. In this work, the authors show how the 2-dimensional interlaced sampling scheme can be exploited to derive a wavelet decomposition using Kronecker products. Their method is an extension of the 1-dimensional non-uniform

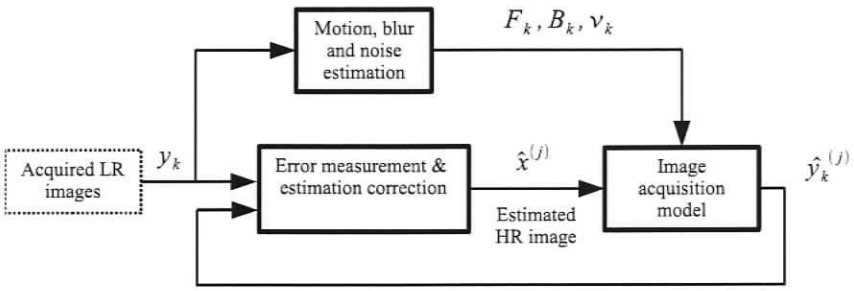


Figure 2.5: Simulate and correct scheme.

interpolation method due to Ford and Etter [29] and it has the advantage of being computationally efficient.

2.3.2 Simulate-and-Correct Methods

Simulate-and-correct methods work in a very intuitive manner. First, it is assumed that the mechanism by which the LR images are obtained is fairly well known. This means that the PSF of the acquisition system, the relative motion between LR frames, and the noise characteristics are known to some extent. Next, a set of LR images is simulated by feeding the observation model with a coarse estimation of the HR image. After that, the HR estimation is corrected according to the differences observed between the simulated and observed LR images. This process is repeated iteratively until some error measure between the observed and simulated images is minimized. A graphical representation of the simulate-and-correct process is shown in Fig. 2.5.

Simulate-and-correct methods differ from one to the other in the error measure and correction strategy chosen. One of the most simple approaches is due to Keren et al. [25]. In this paper, the simulate-and-correct method is used to deblur an HR estimation obtained by some sort of non-uniform interpolation (see Section 2.3.1).

Then, the error measure for the jth iteration is defined as

$$\epsilon^j = \sum_{k=1}^{K} \|\hat{\boldsymbol{y}}_{\boldsymbol{k}}{}^{(j)} - \boldsymbol{y_k}\|_1 \tag{2.11}$$

where $\hat{\boldsymbol{y}}_{\boldsymbol{k}}{}^{(j)}$ is the kth LR image obtained by feeding the observation model with the jth HR estimation, $\boldsymbol{x}^{(j)}$, and $\boldsymbol{y_k}$ is the kth observed LR image. Subsequently, each pixel of $\boldsymbol{x}^{(j)}$ is corrected according to the following scheme: its value is first incremented by a predefined amount and the resulting error, ϵ^{j+}, is computed. Then, the same pixel value is decremented by a predefined amount and similarly ϵ^{j-} is calculated. The corrected value for the considered pixel is the one that leads to the lowest error provided it is lower than ϵ^j. This scheme is repeated for every pixel of $\boldsymbol{x}^{(j)}$ to calculate the corrected estimation $\boldsymbol{x}^{(j+1)}$.

Even though simple and intuitive, the later scheme is quite slow and likely to converge to a local minimizer. An improved version of the described scheme is provided in [30] where the error measure is redefined as

$$\epsilon^j = \sqrt{\sum_{k=1}^{K} \|\hat{\boldsymbol{y}}_{\boldsymbol{k}}{}^{(j)} - \boldsymbol{y_k}\|_2^2} \tag{2.12}$$

and the correction formula for each HR pixel is

$$\begin{aligned} \hat{\boldsymbol{x}}^{(j+1)}{}_{\{x_1,x_2\}} &= \hat{\boldsymbol{x}}^{(j)}{}_{\{x_1,x_2\}} \\ &\quad + \sum_{k=1}^{K} \sum_{Y} \left(\boldsymbol{y_k}{}_{\{y_1,y_2\}} - \hat{\boldsymbol{y}}_{\boldsymbol{k}}{}^{(j)}{}_{\{y_1,y_2\}} \right) h_k^{BP}(x_1, y_1, x_2, y_2) \end{aligned} \tag{2.13}$$

where $\{x_1, x_2\}$ are the coordinates of an HR pixel, $\{y_1, y_2\}$ are the coordinates of an LR pixel, Y is the set of LR coordinates $\{y_1, y_2\}$ influenced by an HR pixel located at $\{x_1, x_2\}$, and h_k^{BP} is referred to as the 'back-projection kernel'. The latter entity measures the influence of each HR pixel on each LR pixel observed. To construct the back-projection kernel, the relative motion between frames and the PSF of the system

need to be specified. An important difference in this correction scheme is that each pixel is corrected according to the influence it has on the current LR pixel difference being measured.

2.3.3 Stochastic Methods

The 'maximum-a-posteriori' (MAP) and a special case of MAP, the 'maximum-likelihood' (ML) methods, are the most prominent of this type of reconstruction methods. In the MAP method both the noise and the image to be reconstructed are regarded to be stochastic in nature. A distinctive feature of this method is that previous information about the desired solution can be introduced in terms of the concept of prior probability distribution which is commonly referred to as 'prior' in the literature [31]. A prior is a marginal probability density function that represents what we know about a certain random variable without having observed it. In the context of superresolution, the random variable is the high-resolution image being estimated. Consequently, a prior is a distribution that will assign a likelihood to each and every high-resolution image pertaining to the solution space. When no prior information is introduced, that is, when the statistical prior is a constant equal to 1, the MAP reconstruction reduces to the ML method.

Considering the image formation model and using stacked notation for (1.1), the observed images can be written as

$$\begin{pmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \vdots \\ \boldsymbol{y_p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{DB_1F_1} \\ \boldsymbol{DB_2F_2} \\ \vdots \\ \boldsymbol{DB_pF_p} \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} \boldsymbol{\nu_1} \\ \boldsymbol{\nu_2} \\ \vdots \\ \boldsymbol{\nu_p} \end{pmatrix} \Rightarrow \boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{\nu} \tag{2.14}$$

MAP methods can be used to find the HR image $\boldsymbol{x}$ that is more likely to lead to the observed images $\boldsymbol{y}$. That is, $\hat{\boldsymbol{x}}_{MAP}$ is an HR image $\boldsymbol{x}$ that maximizes the conditional

probability $P(\boldsymbol{x}|\boldsymbol{y})$. By virtue of Bayes theorem, we can write

$$P(\boldsymbol{x}|\boldsymbol{y}) = \frac{P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x})}{P(\boldsymbol{y})} \tag{2.15}$$

Hence

$$\hat{\boldsymbol{x}}_{MAP} = \underset{\boldsymbol{x}}{\operatorname{argmax}} \frac{P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x})}{P(\boldsymbol{y})} \tag{2.16}$$

and, since $P(\boldsymbol{y})$ is not dependant on $\boldsymbol{x}$, then

$$\hat{\boldsymbol{x}}_{MAP} = \underset{\boldsymbol{x}}{\operatorname{argmax}} P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}) \tag{2.17}$$

where $P(\boldsymbol{y}|\boldsymbol{x}) = P(\boldsymbol{\nu})|_{\boldsymbol{\nu}=\boldsymbol{y}-\boldsymbol{H}\boldsymbol{x}}$ is exclusively determined by the noise probability density function and $P(\boldsymbol{x})$ is the statistical prior corresponding to the HR image.

If the HR image is assumed to result from a zero-mean Gaussian random process with autocorrelation matrix $\boldsymbol{Q}$ and the noise is assumed to be additive white Gaussian (AWGN), then the MAP estimator reduces to a regularized minimum mean-square error (MMSE) estimator, that is,

$$\hat{\boldsymbol{x}}_{MAP} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left(\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 + \boldsymbol{x}^T \boldsymbol{Q}^{-1} \boldsymbol{x} \right) \tag{2.18}$$

The unconstrained optimization problem in (2.18) can be solved by using iterative numerical methods like steepest-descent or conjugate-gradient methods [32].

One problem observed when using the Gaussian prior is that the resulting regularization leads to oversmoothed solutions. In other words the term associated with the Gaussian prior penalizes high frequencies in the HR image. This effect is often undesirable as it tends to destroy original high-frequency information associated with edges. A solution to this problem has been proposed by Schultz et al. [33]. In this work, an edge-preserving Huber Markov Gibbs prior model that leads to a convex constrained optimization scheme is proposed.

Notice that when no prior is assumed, that is, when we assume that every possible HR image is equally likely to occur, i.e., $P(\boldsymbol{x}) = 1$, the MAP estimation reduces to the ML case. In such a scheme

$$\hat{\boldsymbol{x}}_{ML} = \underset{\boldsymbol{x}}{\operatorname{argmax}} P(\boldsymbol{y}|\boldsymbol{x}) \tag{2.19}$$

so that $\hat{\boldsymbol{x}}_{ML}$ is the HR, $\boldsymbol{x}$, that makes $\boldsymbol{y}$ most likely to be observed. If AWGN is assumed, then

$$\hat{\boldsymbol{x}}_{ML} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 \tag{2.20}$$

Even though this formulation is simple, the ill-posed nature of the SR problem requires (2.20) to be regularized.

Summing up, stochastic methods are attractive because they fit well in the spatial-domain observation model, previous knowledge of the solution can be introduced while tackling the ill-posedness nature of the SR problem, and the HR image can often be obtained by efficient gradient-descent methods. In addition, the MAP framework has been shown to be appropriate for solving the problem of simultaneous registration and reconstruction of an HR image from several warped, blurred, noisy and subsampled versions [33–35].

2.4 Wavelet-Domain Methods

Wavelet-based multiresolutional analysis has proven useful in the reconstruction of HR images from low-quality LR frames. The capabilities of wavelets to decompose a signal into a coarse content and detail content along with the existence of nearly optimal wavelet-based denoising algorithms have made this approach an attractive SR tool. Next, we provide an example of an algorithm that makes use of the wavelet decomposition to estimate an HR image.

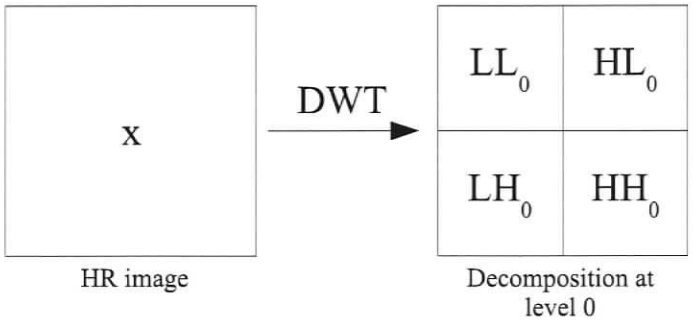


Figure 2.6: DWT of $\boldsymbol{x}$.

2.4.1 Higher-Frequency Subband Estimation

A common feature of higher-frequency subband estimation methods is the assumption that the image to be enhanced is a good approximation of the lowpass-filtered subband of a wavelet-transformed HR image [36]. Another distinctive characteristic of these methods is their capability to estimate higher-frequency content from a single LR image thereby eliminating the need for several views of the scene. This last feature relies on the fact that low-pass wavelet filters are non-ideal thereby allowing high-frequency information to be aliased into low-frequency subbands.

A basic reconstruction scheme is illustrated in Figs. 2.6 and 2.7. In Fig. 2.6, $\boldsymbol{x}$ represents an HR image from which only subband LL_0 is known through the available LR image. By exploiting certain properties of the discrete wavelet transform (DWT) like persistency of the coefficient magnitudes through different resolution scales [36] or local extrema propagation across scales [37], detail coefficients can be estimated. Finally, the estimated coefficients and the original low-frequency subband information can be combined to reconstruct an HR version of the original image. An outline of the process is illustrated in Fig. 2.7.

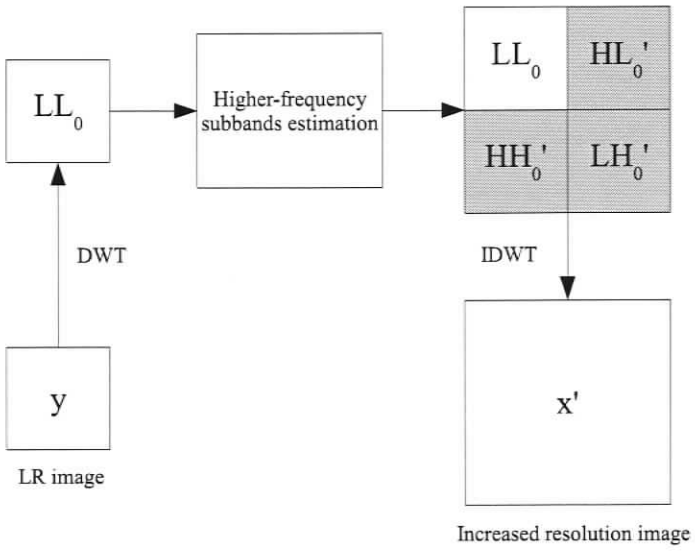


Figure 2.7: Basic reconstruction scheme.

2.5 Summary

In this chapter, we presented an overview of the most representative super-resolution methods. We divided these into three groups, those based on the frequency domain, those based on the spatial domain, and those based on the wavelet domain. Frequency-domain-based methods are of simple formulation and are computationally efficient but fail to incorporate complex motion models, space-dependent degradations, and previous knowledge about the solution. On the other hand, spatial-domain methods, while computationally more expensive, tackle the deficiencies of frequency-domain methods and, at the same time, can incorporate spatially dependent regularizing functionals that improve their numerical conditioning thereby improving their robustness. Finally, we presented some wavelet-based algorithms which can produce an image of increased resolution from a single image without requiring several sub-pixel displaced images.

In the next chapter, we study two state-of-the-art spatial-domain-based algorithms for multiframe super-resolution, give details on their implementations, and provide suggestions for their improvement.

Chapter 3

Study of Two State-of-the-Art Algorithms for Image Super-Resolution

3.1 Introduction

In this chapter, we review two state-of-the-art algorithms for multiframe image super-resolution reported in [9] and [10] by Farsiu et al. These address the problem of robust grayscale super-resolution and the problem of multiframe demosaicing and color super-resolution in the context of spatial-domain-based maximum-likelihood (ML) estimation. First, we provide a description of a procedure to generate artificially degraded images that will serve as inputs for the super-resolution algorithms. Next, we outline the fundamental concepts behind these two algorithms, give details on their implementation, show the results of applying these to a series of artificially degraded images, and provide suggestions for their improvement.

3.2 Generation of Artificially Degraded Images

To evaluate the performance of super-resolution algorithms, a series of images that have undergone controlled degradation is needed. For this reason, a degradation model that emulates the acquisition of a set of N different images for the same scene, such as the one described in Section 1.2.2, has been implemented.

A high-resolution (HR) image, represented by vector $\boldsymbol{x}$, corresponds to the input scene. This image represents the ground truth and, ultimately, it constitutes the data to be estimated by the reconstruction algorithm. The image acquisition process is emulated by warping, blurring, downsampling, and degrading the HR input image with additive noise. By applying the process described N times using different warpings to emulate camera movement, a set of N $(M \times L)$-pixel low-resolution (LR) images, represented by $\{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_N}\}$ is obtained.

The process by which the set of N LR images is produced can be modeled mathematically as [9, 10, 32, 38–40]

$$\boldsymbol{y_k} = \boldsymbol{D}\boldsymbol{H_k}\boldsymbol{F_k}\boldsymbol{x} + \boldsymbol{v_k} \quad k = 1, \ldots, N \tag{3.1}$$

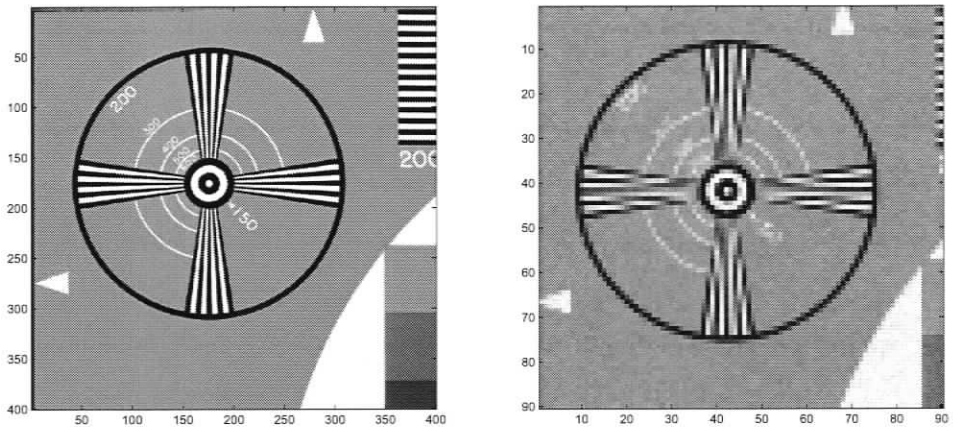
where $\boldsymbol{x}$ represents an r^2ML-element column vector representing a raster scan of the HR image, $\boldsymbol{F_k}$ is an $r^2ML \times r^2ML$ matrix that models the camera motion corresponding to each acquired frame, $\boldsymbol{H_k}$ is an $r^2ML \times r^2ML$ matrix that models the point-spread function (PSF) of the camera, $\boldsymbol{D}$ is an $ML \times r^2ML$ matrix that represents a decimation operation which introduces aliasing artifacts when the resulting resolution is not high enough to resolve the fine details of the scene, and $\boldsymbol{v_k}$ and $\boldsymbol{y_k}$ are ML-element column vectors representing an additive noise source and the observed frame, respectively.

An important simplification to this model can be made if space and time invariant blur is assumed. Space-invariant blur allows $\boldsymbol{F_k}$ and $\boldsymbol{H_k}$ to commute and

time-invariant blur renders $\boldsymbol{H_k} = \boldsymbol{H}$, i.e., the PSF of the camera becomes frame independent. With these assumptions, (3.1) simplifies to

$$\boldsymbol{y_k} = \boldsymbol{DF_kHx} + \boldsymbol{v_k} \quad k = 1, \ldots, N \tag{3.2}$$

The effect of the application of the proposed degradation model to standard image 'EIA1956' [41] shown in Fig. 3.1(a) is illustrated in Fig. 3.1(b).



(a) (b)

Figure 3.1: (a) Original image. (b) Sample degraded image (r = 4, SNR = 18 dB).

It should be mentioned at this point that throughout this work matrices $\boldsymbol{D}$, $\boldsymbol{H}$, and $\boldsymbol{F_k}$ are not constructed. Instead their effects are simulated through direct transformations as suggested in [9] and [10].This results in a much faster and memory-efficient approach.

3.3 An Algorithm for Robust Grayscale Multiframe Super-Resolution

In [9] an algorithm for robust grayscale multiframe super-resolution is presented. The proposed method is based on L_1-norm and robust regularization minimization

which is carried out using a steepest-descent algorithm with a fixed step-size factor. It is claimed in [9] that this approach is not only fast but also robust in that it can effectively cope with non-Gaussian noise and affine (non-purely-translational) camera motion.

3.3.1 Algorithm Description

In what follows we describe and briefly analyze the algorithm proposed in [9]. First, we detail a robust data fusion process known as *non-iterative shift and add* by which a blurred version of the ideal HR image is quickly estimated. Next, we review an iterative process based on a steepest-descent L_1-norm minimization that allows us to obtain a deblurred estimation of the HR image.

Non-Iterative Shift and Add

In [9] the process of finding a high-resolution estimate of a scene from its corresponding set of blurry, noise contaminated, low-resolution images is based on the theory of maximum likelihood (ML) estimators [42].

As was stated in Section 2.3.3, in the super-resolution context the problem of finding the ML estimator can be thought of as finding the HR image $\boldsymbol{x}$ that makes $\{\boldsymbol{y_1}, \ldots, \boldsymbol{y_N}\}$ most likely to be observed. Such an estimate, designated as $\hat{\boldsymbol{x}}$, can be obtained by solving the unconstrained minimization problem

$$\underset{\boldsymbol{x}}{\text{Minimize}} \left[\sum_{k=1}^{N} \rho(\boldsymbol{y_k}, \boldsymbol{D F_k H x}) \right] \tag{3.3}$$

where $\rho(\cdot)$ represents a distance measure between the observed LR images and the degradation model outputs for a given input HR image $\boldsymbol{x}$. The HR estimate $\hat{\boldsymbol{x}}$ is a value of $\boldsymbol{x}$ that minimizes such a distance.

The formulation of function $\rho(\cdot)$, and therefore the ML estimator, depends on the degradation model. For example, $\rho(\cdot)$ depends on the assumed noise model which is

characterized by the probability distribution of $\boldsymbol{v_k}$. If additive white Gaussian noise is assumed, then, as shown in Appendix A.1, the problem of finding an ML estimate for the HR image can be reformulated in terms of the least-squares problem

$$\underset{\boldsymbol{x}}{\text{Minimize}} \left[\sum_{k=1}^{N} \|\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y_k}\|_2^2 \right] \tag{3.4}$$

Finding an ML estimate through a least-squares minimization has been used in many papers such [32] and [38]. However, Farsiu et al. argue that the least-squares-based ML estimator performs poorly in the presence of outliers. That is, when the LR images are contaminated with non-Gaussian noise or when the warping is not purely translational, the least-squares ML estimate is not a good result.

If we consider the blurred version of the HR image $\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x}$, then the multiframe reconstruction problem of the blurred image can be formulated as

$$\underset{\boldsymbol{z}}{\text{Minimize}} \left[\sum_{k=1}^{N} \|\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}\|_p^p \right] \tag{3.5}$$

As shown in Appendix A.2, the gradient of the p-norm-based objective function is

$$\begin{aligned} \boldsymbol{g}_p &= \frac{d}{d\boldsymbol{z}} \left(\sum_{k=1}^{N} \|\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}\|_p^p \right) \\ &= p \sum_{k=1}^{N} \boldsymbol{F_k^T}\boldsymbol{D^T}\text{sign}(\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}) \odot |\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}|^{p-1} \end{aligned} \tag{3.6}$$

Farsiu et al. claim that when $p = 1$, the HR image estimator in (3.5) achieves the highest robustness in the presence of outliers and on the basis of the following observations they have constructed a fast algorithm:

1. When the minimum is reached, the gradient is zero, that is

$$\boldsymbol{g_1} = \sum_{k=1}^{N} \boldsymbol{F_k^T}\boldsymbol{D^T}\text{sign}(\boldsymbol{D}\boldsymbol{F_k}\hat{\boldsymbol{z}} - \boldsymbol{y_k}) = \boldsymbol{0} \tag{3.7}$$

2. $\boldsymbol{D^T}$, which behaves as an upsampling operator, maps an $(M \times L)$- to an $(rM \times rL)$-pixel image after proper zero padding as shown in Fig. 3.2. Consequently, $\boldsymbol{F_k D^T}$ copies the values from the LR grid to the HR grid after proper shifting, which is the effect of applying $\boldsymbol{F_k}$ and zero padding resulting from the multiplication by $\boldsymbol{D^T}$. Similarly, $\boldsymbol{DF_k}$ copies a selected set of pixels in the HR image back to the LR grid.

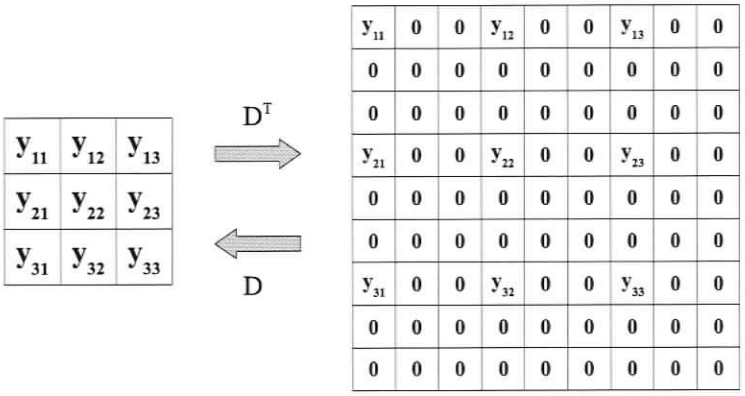


Figure 3.2: Effect of the application of $\boldsymbol{D}$ and $\boldsymbol{D^T}$ matrices. Vectors representing the LR and HR images were rearranged as matrices for easier understanding (r=3).

Under these circumstances, the contribution of the low-resolution frame pixels, $\boldsymbol{y_{k_{ij}}}$, to the computation of each element of $\boldsymbol{g_1}$ assumes one of the following three forms:

1. Addition of a zero resulting from:

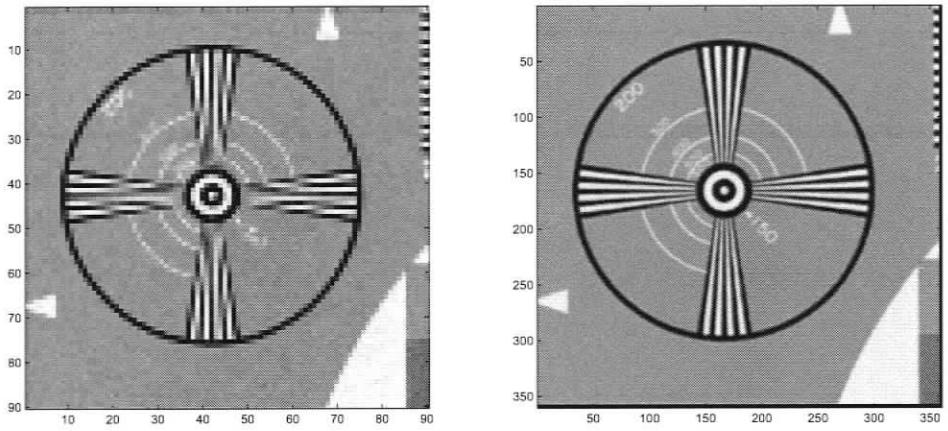
   - zero padding due to upsampling, and
   - $\boldsymbol{y_{k_{ij}}}$ pixels such that $(\boldsymbol{DF_k\hat{z}})_{ij} = \boldsymbol{y_{k_{ij}}}$.

2. Addition of $+1$ due to pixels in $\hat{\boldsymbol{z}}$ larger than the corresponding pixels in frame $\boldsymbol{y_k}$, i.e., $(\boldsymbol{DF_k\hat{z}})_{ij} > \boldsymbol{y_{k_{ij}}}$.

3. Addition of -1 due to pixels in $\hat{\boldsymbol{z}}$ smaller than the corresponding pixels in frame $\boldsymbol{y_k}$, i.e., $(\boldsymbol{DF_k}\hat{\boldsymbol{z}})_{ij} < \boldsymbol{y_{k_{ij}}}$.

As a zero gradient is the result of adding an equal number of $+1s$ and $-1s$, each element of $\hat{\boldsymbol{z}}$ should be the median value of the corresponding elements in the low-resolution frames. The final superresolved image, $\hat{\boldsymbol{x}}$, is calculated by deblurring $\hat{\boldsymbol{z}}$.

This fast approach was implemented and Fig. 3.3(b) shows the image obtained. Note how the aliasing artifacts seen on the black and white stripes were successfully removed.



(a) (b)

Figure 3.3: (a) Sample LR image. (b) Result after applying 'shift and add': A blurred version of the HR frame being estimated.

As suggested by Farsiu et al., the output of this non-iterative algorithm can be later used as the initial estimate, $\hat{\boldsymbol{x}}_0$, for the iterative reconstruction.

Iterative Reconstruction

The multiframe SR problem entails estimating a high-quality HR representation of the observed scene, $\boldsymbol{x}$, from the observed LR images, $\boldsymbol{y_k}$. For this purpose, Farsiu et al., define an objective function $f(\boldsymbol{x}) = J(\boldsymbol{x}) + \lambda R(\boldsymbol{x})$ where $J(\boldsymbol{x})$ and $\lambda R(\boldsymbol{x})$ are a data fidelity term and a regularization term, respectively. The first term assures fidelity or closeness of the final solution to the observed data and the second term mitigates the effect of noise amplification inherent to the inversion of an ill-conditioned system. At the same time, this last term promotes convergence to an edge-preserving solution. Vector $\boldsymbol{x}$ is estimated by solving the optimization problem

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left[J(\boldsymbol{x}) + \lambda R(\boldsymbol{x}) \right] \tag{3.8}$$

Exploiting the robustness of the L_1 norm in the presence of outliers while proposing a regularization term called *bilateral total variation*, a generalization of the total variation principle due to Rudin, Osher, and Fatemi [43], the data fidelity and regularization terms are defined as

$$J(\boldsymbol{x}) = \sum_{k=1}^{N} \| \boldsymbol{D H F_k x} - \boldsymbol{y_k} \|_1 \tag{3.9}$$

and

$$R(\boldsymbol{x}) = \underbrace{\sum_{l=-P}^{P} \sum_{m=0}^{P}}_{l+m \geq 0} \alpha^{|m|+|l|} \| \boldsymbol{x} - \boldsymbol{S_x^l S_y^m x} \|_1 \tag{3.10}$$

where $\boldsymbol{S_x^l}$ and $\boldsymbol{S_y^m}$ are operators that shift the image horizontally and vertically by l and m pixels, respectively.

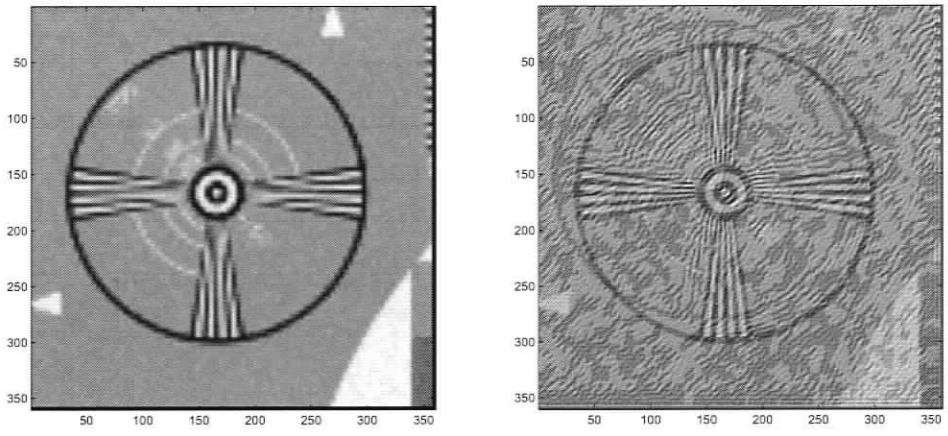
The iterative solution proposed in [9] for the minimization of the problem in (3.8) relies on the following gradient-descent updating scheme

$$\begin{aligned}\hat{\boldsymbol{x}}_{n+1} &= \hat{\boldsymbol{x}}_n - \beta \nabla f(\boldsymbol{x}) \\ \hat{\boldsymbol{x}}_{n+1} &= \hat{\boldsymbol{x}}_n - \beta \sum_{k=1}^{N} \boldsymbol{F_k^T H^T D^T} \text{sign}(\boldsymbol{DHF_k}\hat{\boldsymbol{x}}_n - \boldsymbol{y_k}) \\ &\quad + \beta\lambda \underbrace{\sum_{l=-P}^{P} \sum_{m=0}^{P}}_{l+m\geq 0} \alpha^{|m|+|l|} [\boldsymbol{I} - \boldsymbol{S_y^{-m} S_x^{-l}}] \text{sign}(\hat{\boldsymbol{x}}_n - \boldsymbol{S_x^l S_y^m} \hat{\boldsymbol{x}}_n) \end{aligned} \quad (3.11)$$

Note that constant β controls the step size and that a decrease in the objective function is not guaranteed even if the step is taken in the direction of steepest descent. To illustrate this point, the iterative restoration algorithm was implemented using the parameter values $\lambda = 0.005, P = 2, \beta = 110$, and $\alpha = 0.6$. Figs. 3.4(a) and 3.4(b) show the initial HR image, $\hat{\boldsymbol{x}}_{\boldsymbol{0}}$, generated by a simple bilinear interpolation of one LR frame, and the obtained HR image, respectively. The poor quality of the HR image obtained after several algorithm iterations is evident. Fig. 3.5 shows normalized values for the objective function and gradient norm. As can be noticed, the restoration algorithm has increased the objective function value, that is, the algorithm provides an HR image of lower quality than the initial estimate.

From the results presented in Figs. 3.4 and 3.5, we observe one important shortcoming of this algorithm. Not only an enhancement in the image is not guaranteed but significant degradation may result if the optimization parameters are not chosen properly. A more conservative approach would be to take shorter steps (lower values for β) to ensure a reduction in the objective function in each step; however, this would obviously result in slow progress in finding the minimizer.

The behavior of the algorithm for a small value for β (say, $\beta = 1$) is illustrated in Figs. 3.6 and 3.7. As can be seen in Fig. 3.7, a reduction in the objective function slightly higher than 10% of the initial value is achieved after 10 iterations.



(a) (b)

Figure 3.4: (a) Initial estimate of the HR image, $\hat{\boldsymbol{x}}_{\mathbf{0}}$. (b) Image obtained after 10 iterations.

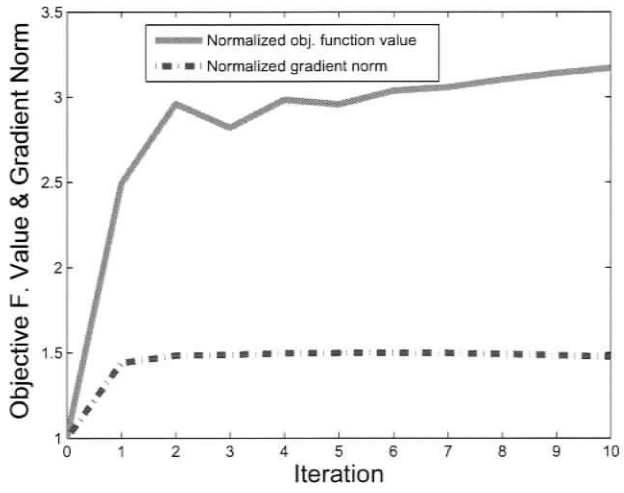
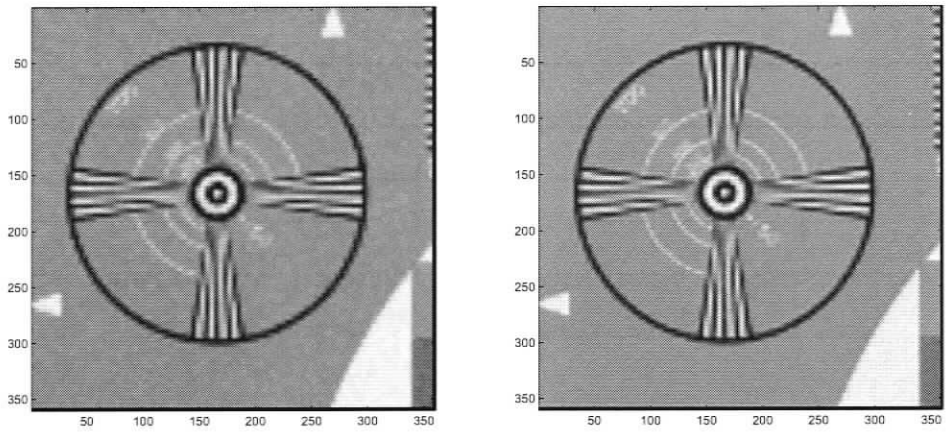


Figure 3.5: Objective function and gradient norm. (Values are normalized relative to the initial ones.)



(a) (b)

Figure 3.6: (a) Initial estimate of the HR image, $\hat{\boldsymbol{x}}_0$. (b) Image obtained after 10 iterations.

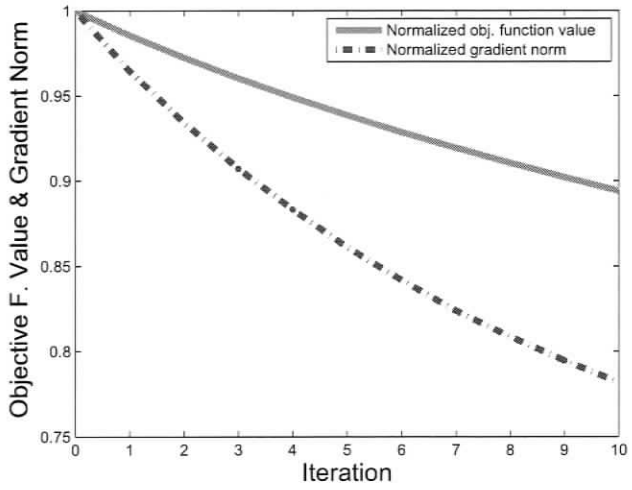


Figure 3.7: Objective function and gradient norm values when $\beta = 1$. (Values are normalized relative to the initial ones.)

The preceding results suggest that the choice of the step-size factor has a significant impact on the quality of the obtained images and the convergence speed.

3.4 An Algorithm for Robust Multiframe Demosaicing and Super-Resolution of Color Images

Most of the research in super-resolution deals with monochromatic images. A rudimentary approach to color super-resolution is to reconstruct each of the primary components separately using a monochrome algorithm [44, 45]. However, these approaches are inherently sub-optimal in that they do not take into account the existing correlation among the different color components.

Color filtering is a process by which each pixel of a sensor array is made sensitive to a particular color band by placing a filter in front of the array. Consequently, to fully characterize the color of a pixel it is necessary to estimate the color information lost in the filtering process. This procedure is known as *demosaicing*. Again, a simple super-resolution approach is to demosaice the LR images before super-resolving them. However, proceeding in this manner prevents us from de-aliasing the color components that are severely undersampled (usually the red and blue components as we are going to show later).

In this section, we study an algorithm for robust multiframe demosaicing and super-resolution of color images reported in [10] by Farsiu et al. This algorithm successfully addresses the problem of simultaneous demosaicing and color super-resolution in a unified context. Much like the algorithm proposed for grayscale images proposed in [9], the reconstruction of a demosaiced increased-resolution image is approached as the minimization of a multi-term objective function.

Color Filtering and Demosaicing

A color image is typically represented by combining the information of three components. In digital imaging, these usually correspond to the three primary colors of the acquisition system, namely red, green, and blue. In practice, to reduce production costs, each pixel of a sensor array is made sensitive to either red, green, or blue. This is usually achieved by placing a color filter array (CFA) in front of the sensor array. As a consequence, there is a particular sampling pattern for each color and, therefore, there is a particular spatial sampling frequency. The most common CFA pattern used in digital image acquisition is the 'Bayer' CFA [46] illustrated in Fig. 3.8.

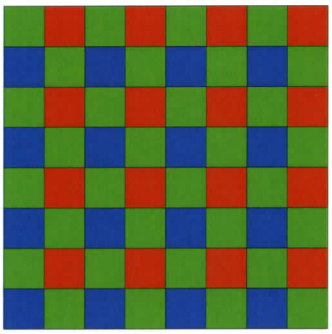
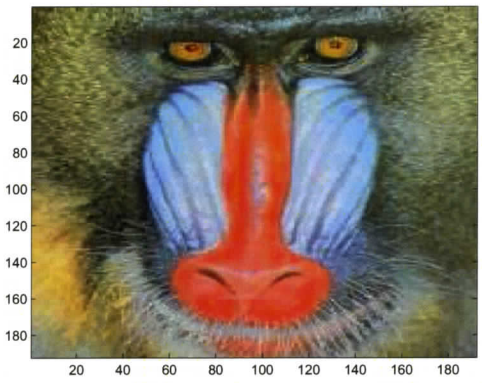


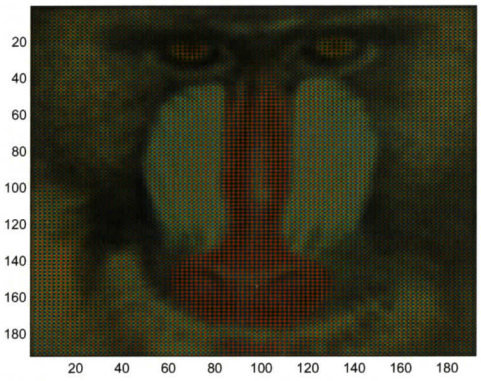
Figure 3.8: Bayer color filter consisting of a 64-pixel (8×8) array with 16 red sensors, 32 green sensors, and 16 blue sensors.

In a Bayer filtered sensor array, 25% of the elements are sensitive to the red colors, 50% are sensitive to the green colors, and the remaining 25% are sensitive to the blue colors. The rationale behind such distribution is that the human visual system is more sensitive to the green band frequencies and, therefore, the spatial sampling frequency for the greens should be made as high as possible. For such a sensor, the resolving power for the green band radiation will be double that of the

red and blue bands. A Bayer-filtered image is shown in Fig. 3.9. Note that in the the color-filtered image each pixel is either red, green, or blue.



(a)



(b)

Figure 3.9: Color filtering. (a) Image 'Baboon'. (b) Color-filtered image 'Baboon'.

To obtain an image that is an accurate representation of the original scene, the values of the missing color components in a color-filtered image must be recovered through demosaicing which is usually achieved by interpolating and upsampling each color component [47, 48].

Acquisition Model for Color-Filtered Images

In a similar manner to that for grayscale images, an image acquisition model is used for color-filtered images and is later inverted to estimate the HR image. The model proposed in [10] is illustrated in Fig. 3.10 and can be represented by the equation

$$\boldsymbol{y_{i,k}} = \boldsymbol{D_{i,k}} \boldsymbol{H} \boldsymbol{F_k} \boldsymbol{x_i} + \boldsymbol{\nu_{i,k}} \quad k = 1, \ldots, N \quad i = R, G, B \tag{3.12}$$

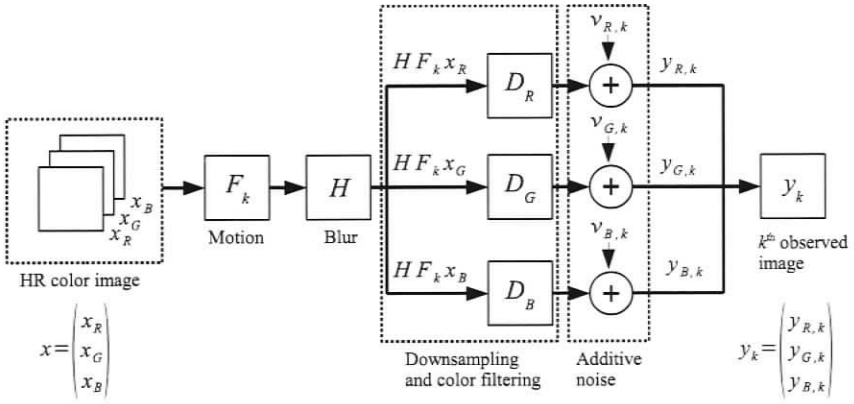


Figure 3.10: Color-filtered image acquisition model.

For simplicity, a square $2rM \times 2rM$-pixel HR color image is assumed where r is the resolution enhancement factor. After lexicographical ordering, each HR component, $\boldsymbol{x_i}$, can be represented by a $4r^2M^2$-element column vector. Thus, a vector representation of the color image, $\boldsymbol{x}$, is achieved by stacking the three components in a $12r^2M^2$-element column vector. Next, this color image is assumed to undergo five effects: camera motion, lens/motion blur, downsampling, color filtering, and additive noise corruption. Since color filtering takes place at the sensor array, all three components are assumed to undergo the same motion and blur effects which are represented by the $4r^2M^2 \times 4r^2M^2$-element matrices $\boldsymbol{F_k}$ and $\boldsymbol{H}$. The downsampling

and color-filtering operations can both be thought of as applied independently to each component and are represented by $M^2 \times 4r^2M^2$-element matrices for the red and blue components, and a $2M^2 \times 4r^2M^2$-element matrix for the green component. Since we are assuming Bayer color filtering, the filtered green component ends up with twice as many pixels than the red and blue components. Additive noise corruption is introduced by adding column vectors $\boldsymbol{\nu_{R,k}}$, $\boldsymbol{\nu_{G,k}}$, $\boldsymbol{\nu_{B,k}}$, which are of length M^2, $2M^2$, and M^2, respectively.

3.4.1 Algorithm Description

In [10] the reconstruction of an HR color image from several undersampled, blurry, color-filtered, and noisy observations is approached as a multi-term MAP-based objective function minimization. This objective function consists of a data fidelity term, two regularization terms operating in the YIQ color space [49], and an intercolor dependencies penalty term defined as follows:

1. *Data Fidelity Term:* This term is designed to assure similarity between the resulting HR image and the observed LR images. Again, an L_1-norm-based term is proposed as

$$J_0(\boldsymbol{x}) = \sum_{i=R,G,B} \sum_{k=1}^{N} \|\boldsymbol{D_i H F_k x_i} - \boldsymbol{y_{i,k}}\|_1 \tag{3.13}$$

   A simplified data fidelity term can be obtained if the blurring is assumed space-invariant. If the latter is true, then $\boldsymbol{F_k}$ and $\boldsymbol{H}$ commute and a penalty term can be defined as

$$J_0(\boldsymbol{x}) = \sum_{i=R,G,B} \|\Phi_i(\boldsymbol{H x_i} - \hat{\boldsymbol{z}}_i)\|_1 \tag{3.14}$$

   where $\hat{\boldsymbol{z}}_i$ are the three color components of the color shift-and-add image $\hat{\boldsymbol{z}}$ obtained with the procedure described in Section 3.3.1. Similarly, matrix $\boldsymbol{\Phi_i}$ is

analogous to matrix $\boldsymbol{A}$ defined for fast grayscale image reconstruction. In other words, $\boldsymbol{\Phi_i}$ is a diagonal matrix with diagonal values equal to the square root of the number of LR pixels that contributed to the measurement of each element of $\hat{\boldsymbol{z}}_{\boldsymbol{i}}$.

2. *Spatial Luminance Regularization:* Given that the human visual system is more sensitive to details in luminance than in chrominance, a bilateral-total-variation regularization term operating on the luminance

$$J_1(\boldsymbol{x}) = \sum_{l=-P}^{P} \sum_{m=-P}^{P} \alpha^{|m|+|l|} \|\boldsymbol{x_L} - \boldsymbol{S_x^l S_y^m x_L}\|_1 \tag{3.15}$$

is introduced, where the luminance is calculated as $\boldsymbol{x_L} = 0.299\boldsymbol{x_R} + 0.587\boldsymbol{x_G} + 0.114\boldsymbol{x_B}$ [49] and matrices $\boldsymbol{S_x^l}$ and $\boldsymbol{S_y^m}$ represent horizontal and vertical shift operators by l and m pixels, respectively.

3. *Spatial Chrominance Regularization:* A less sophisticated regularization for the chrominance information is chosen as the human visual system is less sensitive to chrominance changes. Then, regularization based on the L_2-norm

$$J_2(\boldsymbol{x}) = \|\boldsymbol{\Lambda x_{C1}}\|_2^2 + \|\boldsymbol{\Lambda x_{C2}}\|_2^2 \tag{3.16}$$

is introduced, where $\boldsymbol{x_{C1}}$ and $\boldsymbol{x_{C2}}$ correspond to the I and Q components of the YIQ color representation, that is, $\boldsymbol{x_{C1}} = 0.596\boldsymbol{x_R} - 0.274\boldsymbol{x_G} - 0.322\boldsymbol{x_B}$ and $\boldsymbol{x_{C2}} = 0.211\boldsymbol{x_R} - 0.523\boldsymbol{x_G} + 0.312\boldsymbol{x_B}$ and matrix $\boldsymbol{\Lambda}$ represents a Laplacian high-pass filtering operation. In this way, the regularization term promotes smoothness in the chrominance components of the solution.

4. *Intercolor Dependencies Term:* Even though edges may be defined by larger or smaller gradient magnitudes across the different color components, it is reasonable to assume that their orientations and locations are the same in the

three color bands. Thus, one way to incorporate inter-component correlation information is to enforce the latter assumption by introducing a term that penalizes the differences between edges locations and orientations. Following some work in [47] that suggest that edge location and orientation can be enforced by minimizing the vector product norm, $\|(m_R, m_G, m_B) \times (n_R, n_G, n_B)\|_2^2$, of adjacent pixels, the following penalty term is introduced [10]

$$\begin{aligned} J_3(\boldsymbol{x}) &= \sum_{l=-1}^{1}\sum_{m=-1}^{1} \|\boldsymbol{x_G} \odot \boldsymbol{S_x^l S_y^m x_B} - \boldsymbol{x_B} \odot \boldsymbol{S_x^l S_y^m x_G}\|_2^2 \\ &\quad + \|\boldsymbol{x_B} \odot \boldsymbol{S_x^l S_y^m x_R} - \boldsymbol{x_R} \odot \boldsymbol{S_x^l S_y^m x_B}\|_2^2 \\ &\quad + \|\boldsymbol{x_R} \odot \boldsymbol{S_x^l S_y^m x_G} - \boldsymbol{x_G} \odot \boldsymbol{S_x^l S_y^m x_R}\|_2^2 \end{aligned} \tag{3.17}$$

where $\odot$ is the element-by-element multiplication operator, and $\boldsymbol{S}_x^l$ and $\boldsymbol{S}_y^m$ are the l-pixel-horizontal and m-pixel-vertical shift matrices, respectively.

The iterative reconstruction relies on the following gradient-descent updating scheme

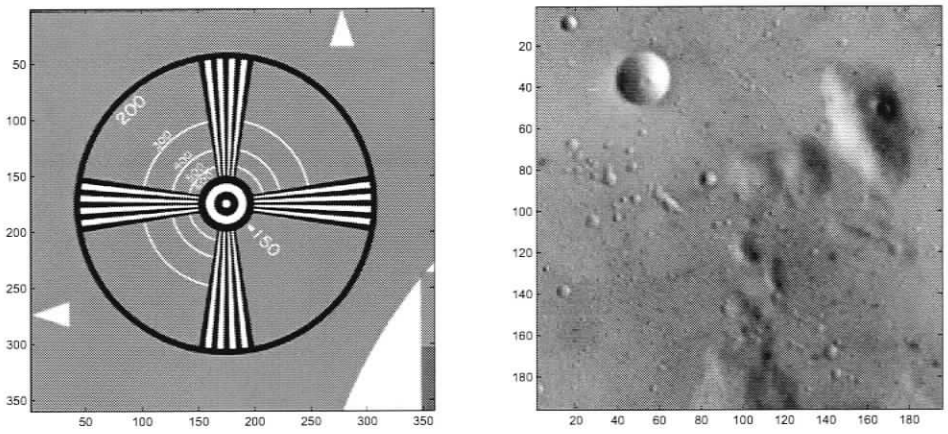
$$\begin{aligned} \hat{\boldsymbol{x}}_{n+1} &= \hat{\boldsymbol{x}}_n - \beta \nabla f(\boldsymbol{x}) \\ \hat{\boldsymbol{x}}_{n+1} &= \hat{\boldsymbol{x}}_n - \beta \left(\nabla J_0(\boldsymbol{x}) + \lambda' \nabla J_1(\boldsymbol{x}) + \lambda'' \nabla J_2(\boldsymbol{x}) + \lambda''' \nabla J_3(\boldsymbol{x})\right) \end{aligned} \tag{3.18}$$

where the scalar factors λ', λ'', and λ''' enable the control of the influence of each term in the objective function and $\nabla J(\boldsymbol{x}_i)$ is the gradient corresponding to each term (see Appendix A.2.2 for the derivation).

3.5 Observations Pertaining to the Steepest-Descent Scheme

The analysis done and results obtained so far reveal that β plays an important role in the performance of the algorithm. It has influence not only on the convergence speed

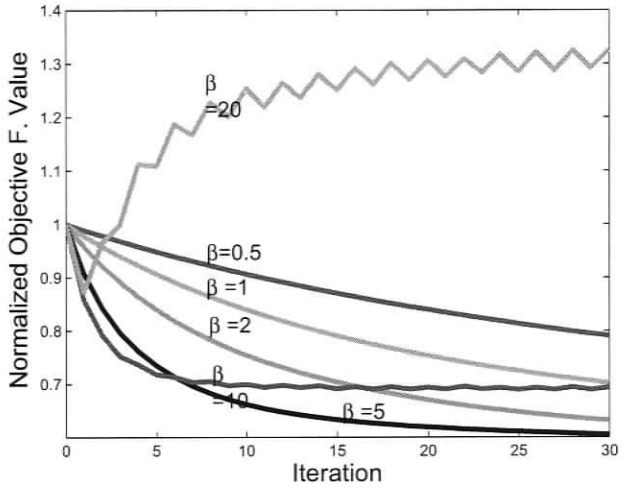
but also on the quality of the images obtained. To illustrate this point, the steepest-descent-based grayscale restoration was run several times, using different values for parameter β, to reconstruct an HR image from a set of artificially degraded images pertaining to image 'EIA1956' shown in Fig. 3.11(a). The values given to β were: $\beta = 0.5, 1, 2, 5, 10, 20$. The results obtained are plotted in Figure 3.12(a). As can be seen, a value $\beta = 0.5$ results in slow improvement of the objective function. Values $\beta = 1$ and $\beta = 10$ lead approximately to the same improvement after 30 iterations although the larger value causes the algorithm to converge faster. Finally, value $\beta = 5$ leads to the best improvement after 30 iterations.



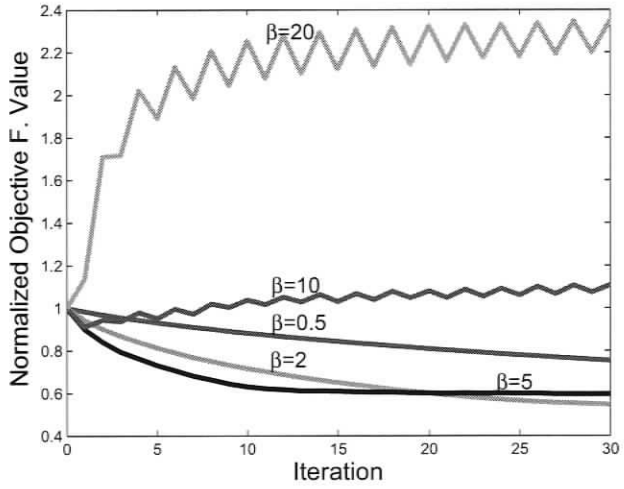
(a) (b)

Figure 3.11: (a) Image 'EIA1956'. (b) Image 'Moon'.

Figure 3.12(b) shows the results of applying the steepest-descent-based restoration with several β values to a set of artificially degraded images pertaining to image 'moon' shown in Fig. 3.11(b). In this case, value $\beta = 10$ no longer improves the objective function and the best improvement after 30 iterations is achieved when $\beta = 2$. We conclude that the choice of a good value β depends on the image being restored.



(a)



(b)

Figure 3.12: Steepest-descent iterative restoration for different values of β. (a) Image 'EIA1956'. (b) Image 'Moon'.

An experiment whereby the value of parameter β was varied was run for the color SR algorithm using image 'house' (Figure 3.13). Plots corresponding to objective function values for different values of the β parameter are shown in Figure 3.14.

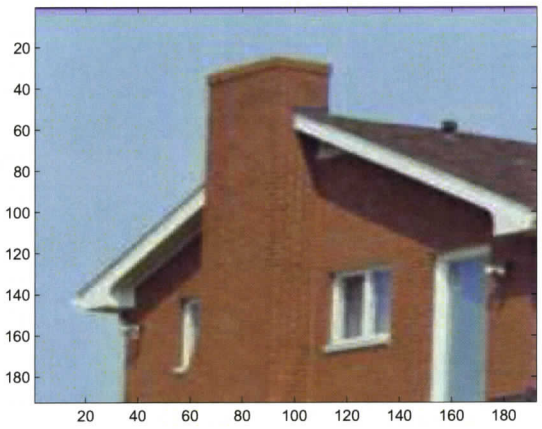


Figure 3.13: Image 'House'.

A desirable property for any algorithm is that a minimum amount of information be required from the user. If an important parameter such as β is left to be provided, then the algorithm performance will depend on the ability of the user to select a proper value. On the other hand, if β is fixed, then, as we have shown in previous sections, the algorithm performance will depend on the specific set of input images. In other words, there is no value of β that would give best results for all possible input images. As a consequence, this parameter needs to be tuned for each case over a large collection of situations.

Another observation that can be made about the steepest-descent scheme has to do with the numerical conditioning of the multiframe image super-resolution problem. It is a widely accepted fact that multiframe image super-resolution is a numerically

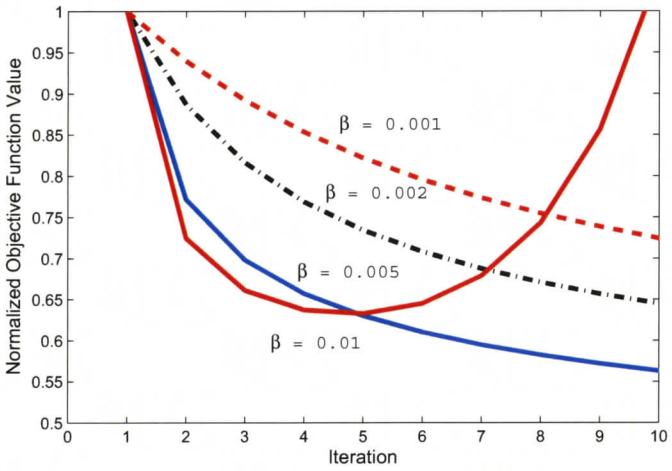


Figure 3.14: Normalized objective function for fixed β (image: 'House').

ill-conditioned problem [39, 50]. Results of an in-depth mathematical study carried out by Wang and Qi [51] point out that the ill-conditioning results from the deblurring part of these algorithms. On the other hand, even though conceptually simple and easy-to-implement, the steepest-descent method is known for its poor convergence properties which are further degraded by numerical ill-conditioning [52, 53].

On the basis of the preceding observations, we conclude that attempting to solve the minimization of the objective functions presented in this chapter by the steepest-descent method with a fixed step-size factor is seriously inefficient.

3.6 Summary

Two state-of-the-art super-resolution algorithms have been presented. Both algorithms are based on the minimization of a multi-term objective function carefully designed to preserve high-frequency information and achieve robustness in the

presence of outliers. However, even though these objective functions were designed taking into account the quality of the images obtained, the proposed minimization approach is quite inefficient. A fundamental tuning parameter is left to be provided by the user, and the iterative approach itself is inefficient due to the ill-conditioned nature of the super-resolution problem and the non-convexity of the color objective function. It is for these reasons that we devote the next two chapters to develop sophisticated algorithms for minimizing the objective functions involved in SR.

Chapter 4

Quasi-Newton Algorithms for Multiframe Image Super-Resolution

4.1 Introduction

In this chapter, we present a much more sophisticated approach for the minimization of the cost functions presented in Chapter 3. In the following sections we present two quasi-Newton algorithms specially reformulated to suit the size of the super-resolution problem. The benefits of adopting these strategies is twofold: on the one hand the minimization algorithm has better convergence properties and is less susceptible to numericall ill-conditioning. On the other hand, the step-size parameter, β, is adaptively calculated by means of an inexact line search (ILS) so that it no longer needs to be selected.

4.2 Fundamentals of Quasi-Newton Minimization

Quasi-Newton methods rank among the most efficient available and, therefore, are used in a wide variety of applications. Like many other optimization techniques,

quasi-Newton methods are developed for the convex quadratic problem are and then extended to the general problem. These methods do not require explicit expressions for the second derivatives of the objective function, they exhibit better convergence properties than the steepest-descent method, and they work well with ILSs [52].

The basic principle in quasi-Newton methods is that the search direction is based on a direction matrix $\boldsymbol{S}$ which serves the same purpose as the inverse Hessian in the Newton method. This matrix is generated from available data and is contrived to be an approximation of the inverse Hessian.

Even though an explicit expression for the Hessian is not required, an estimate of its inverse must be computed, stored, and eventually multiplied by a vector $\boldsymbol{d}$. For the given super-resolution reconstruction problem, the optimal value of an r^2M^2-element vector is sought. Thus, the Hessian of any objective function is an $r^2M^2 \times r^2M^2$ matrix. Traditional quasi-Newton methods, like the Davidon-Fletcher-Powell (DFP) [54,55] or Broyden-Fletcher-Goldfarb-Shanno (BFGS) [56–59] methods, entail computing matrix-vector products involving matrix $\boldsymbol{S}$ in each iteration. Given the considerable size of the Hessian associated with the super-resolution reconstruction problem, turning into a quasi-Newton scheme would, in principle, severely degrade the memory efficiency and involve a great deal of computation.

As an example, the reconstruction of a 256×256-pixel HR image would entail the storage of a $65,536 \times 65,536$ matrix. In addition each iteration would involve a matrix-vector product that would require approximately 4.295×10^9 floating-point multiplications (fpm). To deal with these demands and to be competitive with current super-resolution algorithms, we propose an efficient technique for the storage of $\boldsymbol{S}$ and the computation of the product $\boldsymbol{Sd}$. As will be demonstrated, this technique leads to a significant reduction in memory and computation requirements.

4.3 Application of Quasi-Newton Minimization to Image Super-Resolution

Algorithm 1 shows the structure of a practical quasi-Newton optimization implementation where $f(\boldsymbol{x})$ represents the objective function. The main differences with respect to the steepest-descent scheme implemented in [9] are:

- Instead of fixing a value for β_n, a value that assures reduction in the objective function is sought through an inexact line search such as that due to Fletcher [60].
- The search directions are computed according to a more efficient quasi-Newton approach.

To compute the search directions an estimate of the inverse Hessian, $\boldsymbol{S}_n$, is needed on each iteration. The DFP and BFGS methods provide an iterative means to obtain such estimate through successive rank two corrections. In this way, the inverse Hessian is first approximated by the identity matrix, $\boldsymbol{I}$, and then corrected according to the formulae given in Table 4.1.

For convex quadratic problems, the DFP and BFGS updating formulae entail the following properties [52]:

1. $\boldsymbol{S}_Q$, where Q is the length of vector $\boldsymbol{x}$, is identical to the inverse Hessian.
2. Directions $\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{N-1}$ form a conjugate set.
3. $\boldsymbol{S}_{n+1}$ is positive definite if $\boldsymbol{S}_n$ is positive definite and $\boldsymbol{\delta}_n^T \boldsymbol{\gamma}_n > 0$.

For the convex quadratic case, the first property guarantees finite termination, the second one provides the excellent objective function reduction properties of

Algorithm 1: Practical Quasi-Newton Algorithm

$n = 0$, $\boldsymbol{x}_n = \tilde{\boldsymbol{x}}_0$, $\boldsymbol{g}_n = \nabla f(\tilde{\boldsymbol{x}}_0)$, $\boldsymbol{S}_n = \boldsymbol{I}$;

until $\left|\frac{f(\boldsymbol{x}_n) - f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| < \epsilon_{THR}$ **do**

  $\boldsymbol{d}_n = -\boldsymbol{S}_n \boldsymbol{g}_n$;

  Find an approximate solution to $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$ using an inexact line search;

  $\boldsymbol{\delta}_n = \beta_n \boldsymbol{d}_n$;

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_{n+1})$;

  $\boldsymbol{\gamma}_n = \boldsymbol{g}_{n+1} - \boldsymbol{g}_n$;

  $\boldsymbol{S}_{n+1} = S(\boldsymbol{S}_n, \boldsymbol{\gamma}_n, \boldsymbol{\delta}_n)$;

  $n = n + 1$;

$\boldsymbol{x}^* = \boldsymbol{x}_n$;

Table 4.1: $\boldsymbol{S}_n$ Updating Formulae

Method	$S(\boldsymbol{S}_n, \boldsymbol{\gamma}_n, \boldsymbol{\delta}_n)$
DFP	$\boldsymbol{S}_n + \dfrac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\delta}_n^T \boldsymbol{\gamma}_n} - \dfrac{\boldsymbol{S}_n \boldsymbol{\gamma}_n \boldsymbol{\gamma}_n^T \boldsymbol{S}_n}{\boldsymbol{\gamma}_n^T \boldsymbol{S}_n \boldsymbol{\gamma}_n}$
BFGS	$\boldsymbol{S}_n + \left(1 + \dfrac{\boldsymbol{\gamma}_n^T \boldsymbol{S}_n \boldsymbol{\gamma}_n}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}\right) \dfrac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n} - \dfrac{\left(\boldsymbol{\delta}_n \boldsymbol{\gamma}_n^T \boldsymbol{S}_n + \boldsymbol{S}_n \boldsymbol{\gamma}_n \boldsymbol{\delta}_n^T\right)}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}$

conjugate gradient methods [61], and the third one assures that every step is taken in a descent direction.

By transposition of the formulae given in Table 4.1 it can be shown that if $\boldsymbol{S_n}$ is symmetric then $\boldsymbol{S_{n+1}}$ is also symmetric. On the basis of this observation and letting $\boldsymbol{\mu}_n = \boldsymbol{S}_n \boldsymbol{\gamma}_n$, the updating formulae can be written as in Table 4.2.

4.3.1 Efficient Computation of the Search Directions

As mentioned earlier, unless wisely implemented, quasi-Newton-based image super-resolution may require huge storage and computing capabilities. By letting $Q = r^2 M^2$, we note that the storage of the $\boldsymbol{S}_n$ matrix requires $O(Q^2)$ memory locations and the computation of $\boldsymbol{d}_n = \boldsymbol{S}_n \boldsymbol{g}_n$ requires $O(Q^2)$ fpm. Hence, with Q growing quadratically with the resolution factor and the image size, the storage and computing

Table 4.2: $\boldsymbol{S}_n$ Updating Formulae for $\boldsymbol{S}_0$ Symmetric and $\boldsymbol{\mu}_n = \boldsymbol{S}_n \boldsymbol{\gamma}_n$

Method	$S(\boldsymbol{S}_n, \boldsymbol{\gamma}_n, \boldsymbol{\delta}_n)$
DFP	$\boldsymbol{S}_n + \dfrac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\delta}_n^T \boldsymbol{\gamma}_n} - \dfrac{\boldsymbol{\mu}_n \boldsymbol{\mu}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\mu}_n}$
BFGS	$\boldsymbol{S}_n + \left(1 + \dfrac{\boldsymbol{\gamma}_n^T \boldsymbol{\mu}_n}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}\right) \dfrac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n} - \dfrac{\left(\boldsymbol{\delta}_n \boldsymbol{\mu}_n^T + \boldsymbol{\mu}_n \boldsymbol{\delta}_n^T\right)}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}$

requirements grow very fast. To overcome this difficulty, we propose an efficient way to store matrix $\boldsymbol{S}_n$ and compute $\boldsymbol{S}_n\boldsymbol{g}_n$. For this we find an outer product expansion of $\boldsymbol{S}_n$ and make use of the fact that in each iteration this matrix is corrected by a rank-two matrix to reformulate the expressions in Table 4.2.

In the following two subsections we describe the reformulation of the DFP and BFGS methods.

DFP Method Reformulation

By letting $\zeta_n = (\boldsymbol{\delta}_n^T\boldsymbol{\gamma}_n)^{-1}$, $\tau_n = (\boldsymbol{\gamma}_n^T\boldsymbol{\mu}_n)^{-1}$, and $\boldsymbol{S}_0 = \boldsymbol{I}$, the inverse Hessian update formula can be rewritten as

$$\boldsymbol{S}_{n+1} = \boldsymbol{I} + \sum_{j=0}^{n} \zeta_j \boldsymbol{\delta}_j \boldsymbol{\delta}_j^T - \sum_{j=0}^{n} \tau_j \boldsymbol{\mu}_j \boldsymbol{\mu}_j^T \tag{4.1}$$

where $\boldsymbol{\mu}_0 = \boldsymbol{S}_0\boldsymbol{\gamma}_0 = \boldsymbol{\gamma}_0$, and for $n > 0$

$$\boldsymbol{\mu}_n = \boldsymbol{\gamma}_n + \sum_{j=0}^{n-1} \zeta_j \boldsymbol{\delta}_j \boldsymbol{\delta}_j^T \boldsymbol{\gamma}_n - \sum_{j=0}^{n-1} \tau_j \boldsymbol{\mu}_j \boldsymbol{\mu}_j^T \boldsymbol{\gamma}_n \tag{4.2}$$

Similarly, the search direction vector $\boldsymbol{d}_{n+1}$ can be expressed as

$$\boldsymbol{d}_{n+1} = -\boldsymbol{g}_{n+1} - \sum_{j=0}^{n} \zeta_j \boldsymbol{\delta}_j \boldsymbol{\delta}_j^T \boldsymbol{g}_{n+1} + \sum_{j=0}^{n} \tau_j \boldsymbol{\mu}_j \boldsymbol{\mu}_j^T \boldsymbol{g}_{n+1} \tag{4.3}$$

As can be seen in (4.3), the only information we need to calculate $\boldsymbol{d}_{n+1}$ is $(\zeta_0, \zeta_1, \ldots, \zeta_n)$, $(\tau_0, \tau_1, \ldots, \tau_n)$, $(\boldsymbol{\delta}_0, \boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_n)$, and $(\boldsymbol{\mu}_0, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n)$. In other words, instead of storing a $Q \times Q$ matrix we only store $2(n+1)$ scalars and $2(n+1)$ Q-length vectors. This would result in significant storage savings since the number of iterations is expected to be much smaller than Q.

An algorithm based on the preceding observations is Algorithm 2. In this algorithm, we keep iterating until the stopping criterion

$$\left| \frac{f(\boldsymbol{x_n}) - f(\boldsymbol{x_{n-1}})}{f(\boldsymbol{x_0})} \right| < \epsilon_{THR} \tag{4.4}$$

Algorithm 2: Reformulated DFP Algorithm

$n = 0$, $\boldsymbol{x}_n = \tilde{\boldsymbol{x}}_0$, $\boldsymbol{g}_n = \nabla f(\tilde{\boldsymbol{x}}_0)$, $\boldsymbol{d}_n = -\boldsymbol{g}_n$;

until $\left|\frac{f(\boldsymbol{x}_n) - f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| < \epsilon_{THR}$ **do**

  Find an approximate solution to $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$ using an inexact line search;

  $\boldsymbol{\delta}_n = \beta_n \boldsymbol{d}_n$;

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_{n+1})$;

  $\boldsymbol{\gamma}_n = \boldsymbol{g}_{n+1} - \boldsymbol{g}_n$;

  if $n = 0$ **then** $\boldsymbol{\mu}_n = \boldsymbol{\gamma}_n$, **else** calculate $\boldsymbol{\mu}_n$ according to (4.2);

  $\zeta_n = (\boldsymbol{d}_n^T \boldsymbol{\gamma}_n)^{-1}$;

  $\tau_n = (\boldsymbol{\gamma}_n^T \boldsymbol{\mu}_n)^{-1}$;

  calculate $\boldsymbol{d}_{n+1}$ using (4.3);

  store ζ_n, τ_n, $\boldsymbol{\delta}_n$ and $\boldsymbol{\mu}_n$;

  $n = n + 1$;

$\boldsymbol{x}^* = \boldsymbol{x}_n$;

is met. The idea behind such criterion is to stop the algorithm when the improvement introduced in an iteration is below a predefined threshold. This criterion was found to reduce the number of iterations required in the neighbourhood of the minimizer [52]. We measure this improvement as the ratio of decrease in the objective function value in the last iteration to the objective function value at the beginning of the iterations. Table 4.3 summarizes the storage and computational demands of the proposed reformulated DFP method (Note that the computational effort spent in objective function and objective function gradients evaluation is not being considered).

Table 4.3: Storage and Computational Complexity of Reformulated DFP Method

Parameter	Storage (coefficients)	Computation (fpm)
$\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_n$	$(n+1)Q$	$\approx 4nQ$
$\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_n$	$(n+1)Q$	—
$\zeta_0, \ldots, \zeta_n$	$n+1$	Q
$\tau_0, \ldots, \tau_n$	$n+1$	Q
$\boldsymbol{d}_{n+1}$	Q	$\approx 4(n+1)Q$
Total	$\approx (2n+3)Q$	$\approx (8n+6)Q$

In summary, at the nth iteration $2(n+1)(Q+1) \approx 2Q(n+1)$ coefficients need to be stored. With respect to the computational effort needed to calculate $\boldsymbol{\mu}_n$ and $\boldsymbol{d}_{n+1}$, we observe from (4.2) that to compute each term of the summations, $(2Q+1)$ fpm are required. Consequently, the computation of $\boldsymbol{\mu}_n$ requires $2n(2Q+1)$ fpm. Now, after calculating $\boldsymbol{\mu}_n$, ζ_n, τ_n and using similar arguments as before, we note that the computation of the next search direction $\boldsymbol{d}_{n+1} = \boldsymbol{S}_{n+1}\boldsymbol{g}_{n+1}$ requires approximately $4Q(n+1)$ fpm, thereby totalling approximately $(8n+6)Q$ fpm per iteration. Compared to the original formulation, the new one entails major savings as $n << Q = r^2M^2$.

BFGS Method Reformulation

A similar strategy to the one used for the DFP method leads to an efficient reformulation of the BFGS method. Letting $\zeta_n = (\boldsymbol{\delta}_n^T \boldsymbol{\gamma}_n)^{-1}$, then from Table 4.2

$$\boldsymbol{S}_{n+1} = \boldsymbol{S}_n + \left(1 + \zeta_n \boldsymbol{\gamma}_n^T \boldsymbol{\mu}_n\right) \zeta_n \boldsymbol{\delta}_n \boldsymbol{\delta}_n^T - \zeta_n \left(\boldsymbol{\delta}_n \boldsymbol{\mu}_n^T + \boldsymbol{\mu}_n \boldsymbol{\delta}_n^T\right) \tag{4.5}$$

By letting $\epsilon_n = \left(1 + \zeta_n \boldsymbol{\gamma}_n^T \boldsymbol{\mu}_n\right) \zeta_n$, we have

$$\begin{aligned} \boldsymbol{S}_{n+1} &= \boldsymbol{S}_n + \epsilon_n \boldsymbol{\delta}_n \boldsymbol{\delta}_n^T - \zeta_n \boldsymbol{\delta}_n \boldsymbol{\mu}_n^T - \zeta_n \boldsymbol{\mu}_n \boldsymbol{\delta}_n^T \\ &= \boldsymbol{S}_n + \left(\epsilon_n \boldsymbol{\delta}_n - \zeta_n \boldsymbol{\mu}_n\right) \boldsymbol{\delta}_n^T - \zeta_n \boldsymbol{\delta}_n \boldsymbol{\mu}_n^T \end{aligned} \tag{4.6}$$

and, if we let $\boldsymbol{S}_0 = \boldsymbol{I}$

$$\boldsymbol{S}_{n+1} = \boldsymbol{I} + \sum_{j=0}^{n} \left(\epsilon_j \boldsymbol{\delta}_j - \zeta_j \boldsymbol{\mu}_j\right) \boldsymbol{\delta}_j^T - \sum_{j=0}^{n} \zeta_j \boldsymbol{\delta}_j \boldsymbol{\mu}_j^T \tag{4.7}$$

Now, as in the DFP case $\boldsymbol{\mu}_0 = \boldsymbol{\gamma}_0$ and from (4.7), we have

$$\begin{aligned} \boldsymbol{d}_{n+1} &= -\boldsymbol{S}_{n+1} \boldsymbol{g}_{n+1} \\ &= -\boldsymbol{g}_{n+1} - \sum_{j=0}^{n} \left(\epsilon_j \boldsymbol{\delta}_j - \zeta_j \boldsymbol{\mu}_j\right) \boldsymbol{\delta}_j^T \boldsymbol{g}_{n+1} + \sum_{j=0}^{n} \zeta_j \boldsymbol{\delta}_j \boldsymbol{\mu}_j^T \boldsymbol{g}_{n+1} \end{aligned} \tag{4.8}$$

and for $n > 0$

$$\boldsymbol{\mu}_n = \boldsymbol{S}_n \boldsymbol{\gamma}_n = \boldsymbol{\gamma}_n + \sum_{j=0}^{n-1} \left(\epsilon_j \boldsymbol{\delta}_j - \zeta_j \boldsymbol{\mu}_j\right) \boldsymbol{\delta}_j^T \boldsymbol{\gamma}_n - \sum_{j=0}^{n-1} \zeta_j \boldsymbol{\delta}_j \boldsymbol{\mu}_j^T \boldsymbol{\gamma}_n \tag{4.9}$$

Based on the preceding observations, the BFGS method can be implemented in terms of Algorithm 3.

Table 4.4 summarizes the storage and computational demands of the proposed reformulated BFGS method. At the nth iteration, the storage of approximately $(2n + 3)Q$ coefficients is needed. Following the same reasoning as for the DFP reformulation, we observe that the computation of the next search direction $\boldsymbol{d}_{n+1}$

Algorithm 3: Reformulated BFGS Algorithm

$n = 0$, $\boldsymbol{x}_n = \tilde{\boldsymbol{x}}_0$, $\boldsymbol{g}_n = \nabla f(\tilde{\boldsymbol{x}}_0)$, $\boldsymbol{d}_n = -\boldsymbol{g}_n$;

until $\left|\frac{f(\boldsymbol{x}_n) - f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| < \epsilon_{THR}$ **do**

  Find an approximate solution to $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$ using an inexact line search;

  $\boldsymbol{\delta}_n = \beta_n \boldsymbol{d}_n$;

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_{n+1})$;

  $\boldsymbol{\gamma}_n = \boldsymbol{g}_{n+1} - \boldsymbol{g}_n$;

  if $n = 0$ **then** $\boldsymbol{\mu}_n = \boldsymbol{\gamma}_n$, **else** calculate $\boldsymbol{\mu}_n$ according to (4.9);

  $\zeta_n = (\boldsymbol{d}_n^T \boldsymbol{\gamma}_n)^{-1}$;

  $\epsilon_n = \left(1 + \zeta_n \boldsymbol{\gamma}_n^T \boldsymbol{\mu}_n\right) \zeta_n$;

  calculate $\boldsymbol{d}_{n+1}$ according to (4.8);

  store ζ_n, ϵ_n, $\boldsymbol{\delta}_n$ and $\boldsymbol{\mu}_n$;

  $n = n + 1$;

$\boldsymbol{x}^* = \boldsymbol{x}_n$;

Table 4.4: Storage and Computational Complexity of Reformulated BFGS Method

Parameter	Storage (coefficients)	Computation (fpm)
$\boldsymbol{\mu}_0, \ldots, \boldsymbol{\mu}_n$	$(n+1)Q$	$\approx 5nQ$
$\boldsymbol{\delta}_0, \ldots, \boldsymbol{\delta}_n$	$(n+1)Q$	—
$\zeta_0, \ldots, \zeta_n$	$n+1$	Q
$\epsilon_0, \ldots, \epsilon_n$	$n+1$	Q
$\boldsymbol{d}_{n+1}$	Q	$\approx 5(n+1)Q$
Total	$\approx (2n+3)Q$	$\approx (10n+7)Q$

requires approximately $5Q(2n+1)$ fpm, thereby, totalling approximately $(10n+7)Q$ fpm per iteration.

Table 4.5 summarizes the storage and computation savings achieved for the reconstruction of a 256 × 256-pixel image after 50 iterations. It can be observed that for the worst case, that is, for the last iteration, $\boldsymbol{S}_{n+1}$ requires around 650 times fewer storage locations and the computation of the next search direction, $\boldsymbol{d}_{n+1}$, is approximately 140 times faster than using the original formulation.

Table 4.5: Storage and Computational Requirements Comparison for the Reconstruction of a 256 × 256-pixel Image after 50 Iterations ($n_{final} = 50$).

Method	$\boldsymbol{S}_{n+1}$ storage (coefficients)	$\boldsymbol{d}_{n+1}$*computation* (fpm)
Original formulation	$\approx 4.295 \times 10^9$	$\approx 4.295 \times 10^9$
Reformulated DFP	$\approx 6.684 \times 10^6$	$\approx 26.476 \times 10^6$
Reformulated BFGS	$\approx 6.684 \times 10^6$	$\approx 33.095 \times 10^6$

4.3.2 A Practical Inexact Line Search Algorithm

In the proposed quasi-Newton minimization, as in other multidimensional minimizations, most of the computational effort is spent in objective function and gradient evaluations required by the line search step. As more precision is required for the solution of $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$ more line search iterations will be needed. On the other hand, if precision requirements are relaxed then the computational load can be reduced.

Fortunately, the convergence of the DFP and BFGS methods is assured even if the solution of the line search not exact. This allows us to use an efficient ILS algorithm based on the one proposed by Fletcher [60] to get an approximation to β^*, the minimizer of $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$.

Starting from an initial search interval (β_L, β_U), the ILS algorithm iteratively reduces this interval through quadratic two-point interpolations and extrapolations. The search is stopped when a $\beta = \beta_0$ is found which leads to a prescribed reduction in $f(\boldsymbol{x})$ and $|\frac{d}{d\beta} f(\boldsymbol{x}_n + \beta \boldsymbol{d}_n)|$. These conditions are

$$f(\boldsymbol{x}_n + \beta_0 \boldsymbol{d}_n) < f(\boldsymbol{x}_n) + \rho(\beta_0 - \beta_L) f'_L \tag{4.10}$$

$$|f'(\boldsymbol{x}_n + \beta_0 \boldsymbol{d}_n)| < |f'(\boldsymbol{x}_n)| \tag{4.11}$$

where ρ is a constant ranging from 0 to $\frac{1}{2}$ and f' represents a directional derivative in the search direction, that is,

$$f'(\boldsymbol{x}) = \frac{d}{d\alpha} f(\boldsymbol{x} + \alpha \boldsymbol{d}_n) = \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}_n \tag{4.12}$$

A practical implementation of the inexact line search [52] is summarized in terms of Algorithm 4 . This implementation is based on the one proposed by Fletcher [60] and accounts for several situations that arise in practice when working with finite precision calculations and when minimizing arbitrary objective functions.

In Algorithm 4, ρ and σ are parameters ranging from 0 to $\frac{1}{2}$ and from 0 to 1 respectively. These constants enable the user to adjust the algorithm precision. Parameters τ and χ range from 0 to 1 and control the search interval reduction in each iteration. Constant $\hat{M}$ limits the number of iterations allowed for the line search. Finally, parameter ϵ_2 is a constant whose value is of the order of the roundoff error and serves as a threshold to terminate the search when $f(\boldsymbol{x}_n + \beta \boldsymbol{d}_n)$ has a very shallow minimum with respect to β or does not have a well-defined minimizer.

A practical quasi-Newton algorithm that makes use of an ILS is Algorithm 5. Since a nonpositive definite $\boldsymbol{S_{n+1}}$ can sometimes arise near a solution point due to numerical ill-conditioning, a safeguard to assure that each $\boldsymbol{d}_n$ is a descent direction is implemented in lines 9 to 12. In this way, if $\boldsymbol{S}_{n+1}$ is not positive definite then the identity matrix is assigned to $\boldsymbol{S}_{n+1}$ so that the next step is taken in the steepest-descent direction.

4.4 Results

In this section we present the results of experiments aimed at assessing the benefits of adopting the proposed strategies. We will focus on the improvements in convergence time and quality of the reconstructed images.

4.4.1 Peak Signal-To-Noise Ratio

The mean-squared error (MSE) and its associated quantity, peak signal-to-noise ratio (PSNR), are two widely used full-reference quality metrics because they are easy to calculate and they have clear physical meanings [62]. The MSE is a measure of the power of an additive noise signal while the PSNR is a ratio beetwen the maximum power of a signal and the power of the noise that corrupts it. Then, the PSNR is a measure of the purity of a signal. In the context of super-resolution, if we consider an

Algorithm 4: Practical Inexact Line Search Algorithm

Data: $\boldsymbol{x}_n, f(\boldsymbol{x}_n), \boldsymbol{d}_n, \boldsymbol{g}_n$

Set algorithm parameters: $\rho, \sigma, \tau, \chi, \hat{M}, \epsilon_2$;

$m = 0$, $f_0 = f_{00} = f(\boldsymbol{x}_n)$, $\beta_L = 0$, $\beta_U = \infty$;

if *first call to algorithm* **then** $\Delta f_0 = f_0$

$f_L = f_0$, $f'_L = \boldsymbol{g}_n^T \boldsymbol{d}_n$;

if $|f'_L| > \epsilon_2$ **then** $\beta_0 = -2\frac{\Delta f_0}{f'_L}$ **else** $\beta_0 = 1$

if $\beta_0 > 1$ *OR* $\beta_0 < 0$ **then** $\beta_0 = 1$

$continueSearch = True$;

while *continueSearch* **do**

  while $f(\boldsymbol{x}_n + \beta_0 \boldsymbol{d}_n) \geq f(\boldsymbol{x}_n) + \rho(\beta_0 - \beta_L) f'_L$ *AND continueSearch* **do**

    $m = m + 1$, $\boldsymbol{\delta}_n = \beta_0 \boldsymbol{d}_n$, $f_0 = f(\boldsymbol{x}_n + \boldsymbol{\delta}_n)$;

    if $f_0 > f_L + \rho(\beta_0 - \beta_L) f'_L$ *AND* $|f_L - f_0| > \epsilon_2$ *AND* $m < \hat{M}$ **then**

      if $\beta_0 < \beta_U$ **then** $\beta_U = \beta_0$,

      $\breve{\beta}_0 = \beta_L + \frac{(\beta_0 - \beta_L)^2 f'_L}{2[f_L - f_0 + (\beta_0 - \beta_L) f'_L]}$;

      $\breve{\beta}_0 = \max[\breve{\beta}_0, \beta_L + \tau(\beta_U - \beta_L)]$, $\breve{\beta}_0 = \min[\breve{\beta}_0, \beta_U - \tau(\beta_U - \beta_L)]$, $\beta_0 = \breve{\beta}_0$;

    $continueSearch = False$

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_n + \beta_0 \boldsymbol{d}_n)$, $f'_0 = \boldsymbol{g}_{n+1}^T \boldsymbol{d}_n$;

  if $|f'_0| > |\sigma f'_L|$ *AND* $|f_L - f_0| > \epsilon_2$ *AND* $m < \hat{M}$ **then**

    $continueSearch = True$;

    $\Delta\beta_0 = \frac{(\beta_0 - \beta_L) f'_0}{f'_L - f'_0}$;

    if $\Delta\beta_0 \leq 0$ **then** $\Delta\beta_0 = \beta_0$;

    $\breve{\beta}_0 = \beta_0 + \Delta\beta_0$, $\breve{\beta}_0 = \min[\breve{\beta}_0, \beta_0 + \chi(\beta_U - \beta_0)]$;

    $\beta_L = \beta_0$, $\beta_0 = \breve{\beta}_0$, $f_L = f_0$, $f'_L = f'_0$;

  else

    $continueSearch = False$;

$\Delta f_0 = f_{00} - f_0$, store Δf_0;

$\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

Algorithm 5: Practical Quasi-Newton Algorithm

$n = 0$, $\boldsymbol{x}_n = \tilde{\boldsymbol{x}}_0$, $\boldsymbol{g}_n = \nabla f(\tilde{\boldsymbol{x}}_0)$, $\boldsymbol{S}_n = \boldsymbol{I}$;

until $\left|\frac{f(\boldsymbol{x}_n)-f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| < \epsilon_{THR}$ **do**

  $\boldsymbol{d}_n = -\boldsymbol{S}_n\boldsymbol{g}_n$;

  find an approximate solution to $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n\boldsymbol{d}_n)$ using an inexact line search;

  $\boldsymbol{\delta}_n = \beta_n\boldsymbol{d}_n$;

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_{n+1})$;

  $\boldsymbol{\gamma}_n = \boldsymbol{g}_{n+1} - \boldsymbol{g}_n$;

  if $\boldsymbol{\delta}_n^T\boldsymbol{\gamma}_n \leq 0$ **then**

    $\boldsymbol{S}_{n+1} = \boldsymbol{I}$

  else

    $\boldsymbol{S}_{n+1} = S(\boldsymbol{S}_n, \boldsymbol{\gamma}_n, \boldsymbol{\delta}_n)$

  $n = n + 1$;

  stopping criterion: **if** $\left|\frac{f(\boldsymbol{x}_n)-f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| \leq \zeta$ *OR* $n > \hat{N}$ **then** stop;

$\boldsymbol{x}^* = \boldsymbol{x}_n$;

$M \times N$-pixel image $\boldsymbol{x}$ and its reconstructed version from blurry, undersampled, and noisy frames to be $\hat{\boldsymbol{x}}$, the MSE is

$$\text{MSE} = \frac{1}{MN} \sum_{i=0}^{MN-1} (\hat{\boldsymbol{x}}_{\boldsymbol{i}} - \boldsymbol{x}_{\boldsymbol{i}})^2 \tag{4.13}$$

and the PSNR is

$$\text{PSNR} = 10 \log_{10} \left\{ \frac{[\max(\boldsymbol{x}_i)]^2}{\text{MSE}} \right\} = 20 \log_{10} \left[\frac{\max(\boldsymbol{x}_i)}{\sqrt{\text{MSE}}} \right] \tag{4.14}$$

where $\max(\boldsymbol{x}_i)$ the maximum value that a pixel can have for a given image representation. Then, for a B-bit representation

$$\text{PSNR} = 20 \log_{10} \left(\frac{2^B - 1}{\sqrt{\text{MSE}}} \right) \tag{4.15}$$

and for a floating-point representation in which usually the pixels values range from 0 to 1

$$\text{PSNR} = 20 \log_{10} \left(\frac{1}{\sqrt{\text{MSE}}} \right) \tag{4.16}$$

In the rest of this thesis, we use the PSNR and DPSNR (difference in PSNR) as metrics for image quality and image quality improvement, respectively.

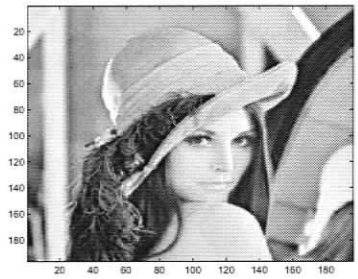
4.4.2 Robust Grayscale Multiframe Super-Resolution

Starting with image 'Lena' [41] (see Fig. 4.1(a)), three low-quality LR sets were generated by applying the degradation model in (3.1) with three different sets of parameters as detailed in Table 4.6. Subsequently, each LR set was used to obtain a higher-resolution image with the SD and quasi-Newton-based algorithms. The same procedure was repeated with images 'Moon' and 'Baboon' [41] (see Figs. 4.1(b) and 4.1(c)). The same stopping criterion was used in every case. Iterations were stopped

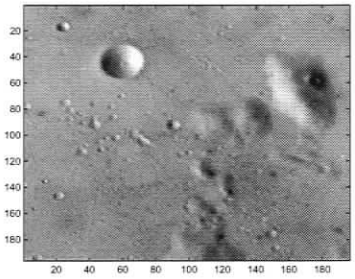
when the reduction in the objective function was below a specified threshold, that is, when

$$\left| \frac{f(\boldsymbol{x_n}) - f(\boldsymbol{x_{n-1}})}{f(\boldsymbol{x_0})} \right| < \epsilon_{THR} \tag{4.17}$$

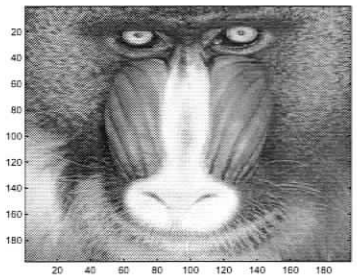
We found that a value of $\epsilon_{THR} = 10^{-3}$ gives good results.



(a)



(b)



(c)

Figure 4.1: Images used for experiment: (a) 'Lena' (b) 'Moon' (c) 'Baboon'.

The performance of the SD-based reconstruction is strongly influenced by the choice of β as it has impact not only on the convergence speed but also on the quality of the reconstructed image. The optimum value of β is not known *a priori* and, furthermore, it depends on the LR image set. Therefore, for the comparison to be

Table 4.6: LR Image-Set Specifications [1]

LR Set Type	SNR (dB)	Blur σ (pixels)
a	24	1
b	24	3
c	18	1

[1] $N = 16$, images were obtained by applying 16 different translational warpings, 3×3 pixels Gaussian blur, a downsample factor of 4, and additive Gaussian noise.

realistic, each image was reconstructed by the SD-based algorithm 100 times using β values uniformly randomized between 0.5 and 10. Then average performance statistics were obtained.

The results obtained for the SD, DFP, and BFGS algorithms are given in terms of bar plots in Figs. 4.2, 4.3, and 4.4. As can be seen in Figs. 4.3 and 4.4, the proposed algorithms lead to a significant reduction in the amount of computation required. In Fig. 4.2 we note that the image quality achieved with quasi-Newton method is slightly superior to the one achieved with SD. This suggests that the quasi-Newton algorithms converge somewhat closer to the minimizer.

Also we observe that the BFGS algorithms outperformed the DFP algorithms in almost every case simulated and for almost every performance measure, which is consistent with Fletcher's simulation results reported in [60].

Figure 4.5 shows sample images from the three different LR images sets generated using image 'Lena' and their corresponding reconstructed HR images using the BFGS algorithms.

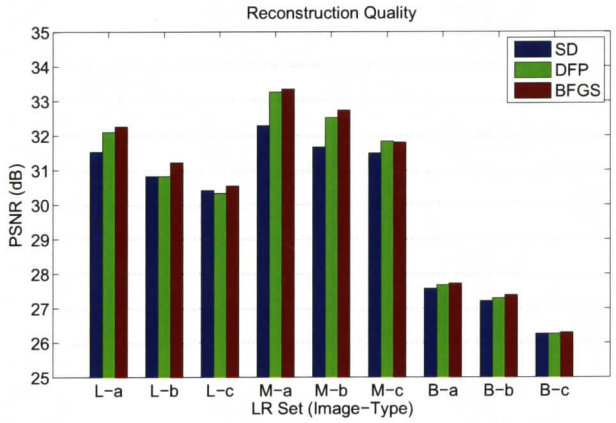


Figure 4.2: Comparative plot for image quality.

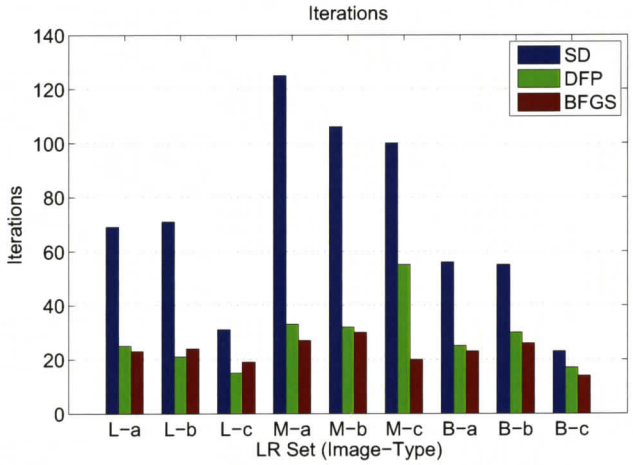


Figure 4.3: Comparative plot for iterations until convergence.

4.4.3 Robust Multiframe Demosaicing and Super-Resolution

In this section, we summarize the results of experiments aimed at assessing the performance of the proposed quasi-Newton algorithms when applied to the problem of simultaneous demosaicing and color super-resolution. As in the previous section, we generated 12 sets of degraded and color-filtered images by applying 4 different sets

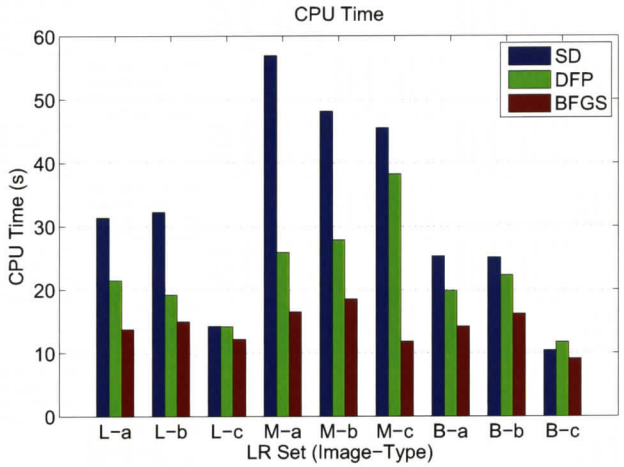


Figure 4.4: Comparative plot for CPU time until convergence.

of degradations, as specified in Table 4.7, to the three color images shown in Fig. 4.6. Then, from each LR set a demosaiced HR image was reconstructed by means of 4 different methods, all based on the cost function described in Section 3.4.1:

1. The first method consisted of applying the iterative scheme proposed in [10], i.e. , fixed step-size factor steepest-descent. We ran this experiment 100 times using a randomized β value.

2. For the second method we approached the reconstruction by implementing a steepest-descent algorithm for which the step-size factor was adaptively adjusted on each iteration by means of the inexact line search described in this Section 4.3.2. The idea behind this experiment is to assess the importance of the search direction in the minimization scheme.

3. A third method consisted in applying the reformulated DFP method implemented by Algorithm 2.

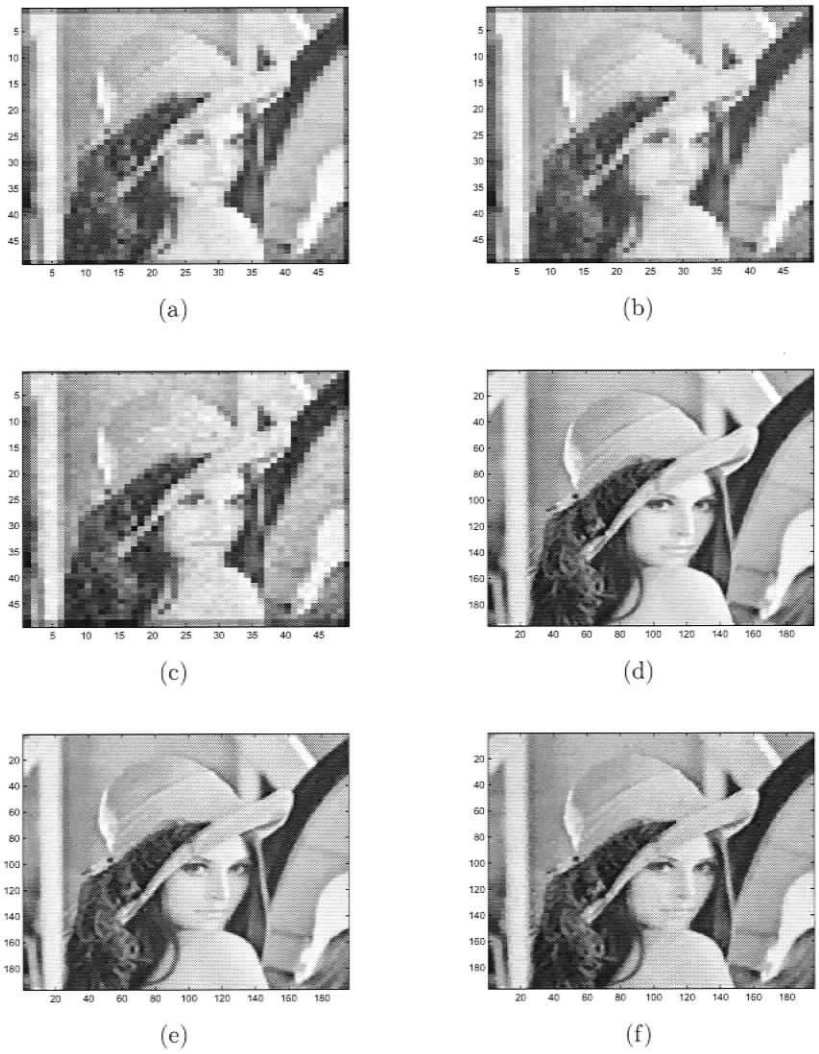


Figure 4.5: Sample LR images and their corresponding reconstructions using BFGS: (a) 'Lena' LR - a. (b) 'Lena' LR - b. (c) 'Lena' LR - c. (d) Reconstructed 'Lena' - a. (e) Reconstructed 'Lena' - b. (f) Reconstructed 'Lena' - c

Table 4.7: LR Image-Set Specifications [1]

LR Set Type	SNR (dB)	Blur σ (pixels) [2]
a	30	1
b	30	3
c	18	1
d	18	3

[1] $N = 10$, images obtained by applying 10 different translational warpings, Gaussian blur, a downsample factor of 4, and additive Gaussian noise.

[2] Blur kernel size is 5×5 pixels.

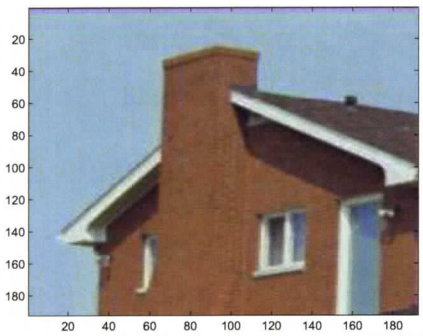
4. Lastly, a fourth method consisted of applying the reformulated BFGS method implemented by Algorithm 3.

In all cases the stopping criterion was the same, iterations were stopped when the reduction in the objective function was below a specified threshold, that is, when

$$\left| \frac{f(\boldsymbol{x_n}) - f(\boldsymbol{x_{n-1}})}{f(\boldsymbol{x_0})} \right| < \epsilon_{THR} \tag{4.18}$$

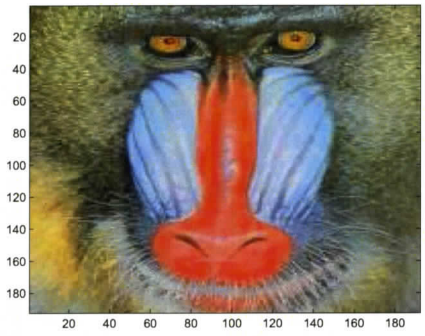
For all cases the initial guess (starting point of the iterations) was a demosaiced version obtained by means of bilinear interpolation of the shift-and-add result of the input LR frames.

In Fig. 4.7, we show a comparative bar plot illustrating the image quality obtained with each algorithm as measured by the PSNR. In this plot we show the mean performance and the best performance, i.e., the one that achieved highest PSNR from the 100 runs of fixed step-size steepest-descent (SD). We also show the results of the ILS equipped SD, the DFP, and BFGS algorithms. It can be observed that in



(a)

(b)



(c)

Figure 4.6: Images used for the experiments: (a) 'House' (196 × 196 pixels). (b) 'Lena' (196 × 196 pixels). (c) 'Baboon' (196 × 196 pixels).

all cases the proposed quasi-Newton algorithms have achieved a significantly higher PSNR.

The step-size factor that produces the best results is not known *a priori* and searching for it would be quite impractical. A much more realistic measure is in terms of the mean performances. However, the results obtained for the best performance of the fixed step-size factor algorithm offer an interesting piece of information. They

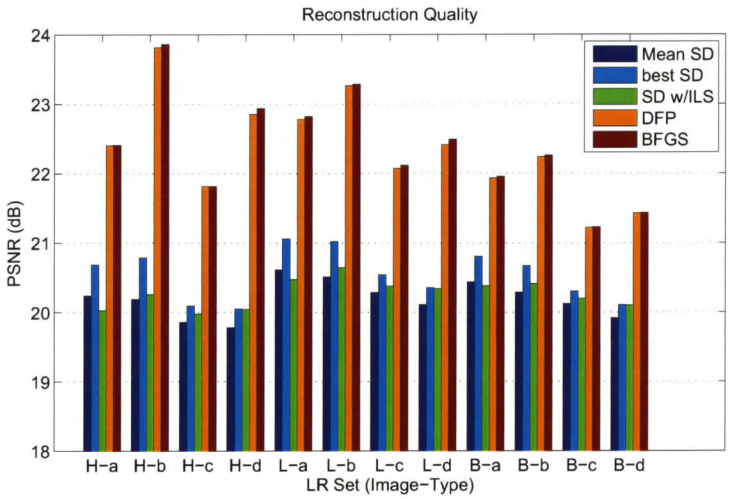


Figure 4.7: Comparative bar plot for image quality (PSNR)

provide a figure of the potential of this algorithm. In other words, they tell us how far we can get with a fixed step-size factor.

Figure 4.8 illustrates how the choice of the step-size factor, β, influences the quality of the reconstructed image. This scatter plot was obtained by reconstructing the LR set 'House-a' with 100 randomized β values and plotting the corresponding PSNR difference (DPSNR) for each step-size factor. From this plot, we also conclude that the range selected for the β randomization was adequate enough to capture the best performance that could be achieved by using a steepest-descent algorithm with a fixed step-size factor.

At this point note that, according to [10], the objective function under consideration is nonconvex. Bearing this in mind, the superior minimization capabilities and therefore quality enhancement capabilities of the proposed algorithms is evident from Fig. 4.9. From this figure we note that the proposed algorithms achieve a better reduction of the objective function and that for every iteration the image quality attained (measured by the PSNR) is superior.

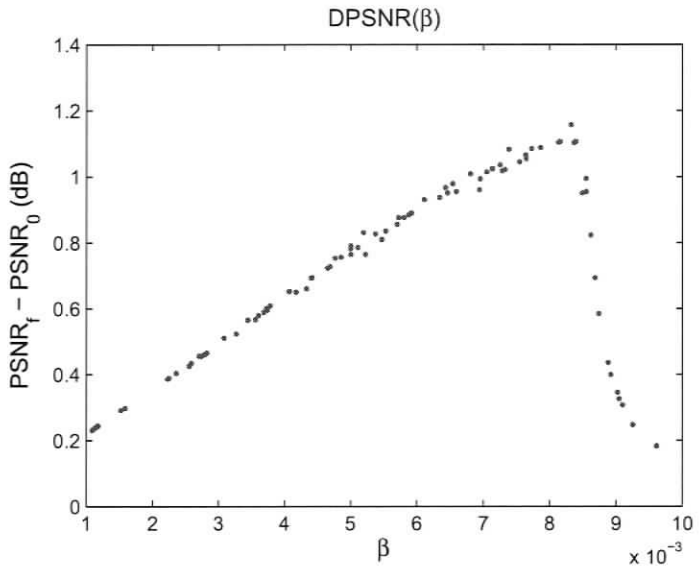
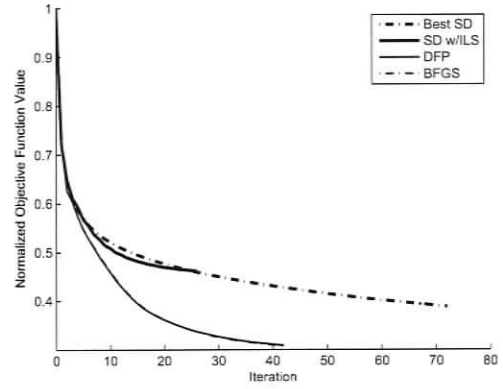


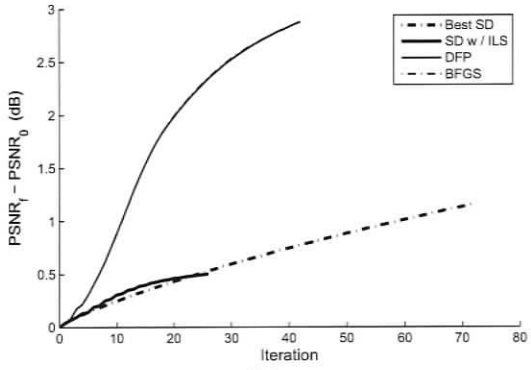
Figure 4.8: DPSNR as a function of the step-size factor pertaining to 'H-a' LR image set reconstruction

Next, in Fig. 4.10 we show the reconstructed images by means of the different algorithms described above for the LR set type 'a' corresponding to image 'House'. From these figures, it can be noticed that the quasi-Newton algorithms lead to cleaner and sharper images. In addition, we observe that an artifact is present in the reconstructed images. Note that the roof edge, magnified in Fig.4.11, shows colored bands where it should be white. This artifact is due to aliasing and is much less pronounced in the images reconstructed by the proposed algorithms. In other words, the proposed algorithms achieve a more accurate demosaicing.

In Figs. 4.12, 4.13, and 4.14, we compare the computational effort in terms of the number of objective function and gradient evaluations and CPU time required for convergence. As expected, quasi-Newton algorithms outperform steepest-descent



(a)



(b)

Figure 4.9: Objective function reduction and PSNR enhancement for the reconstruction of 'House' image from the a-type LR set (a) Objective function evolution. (b) PSNR difference (DPSNR) evolution. NOTE: DFP and BFGS curves are superimposed.

algorithms in terms of the number of objective function and gradient evaluations. However, there is an implicit computation time overhead corresponding to the calculation of the search directions (see Eqs. (4.2), (4.3), (4.9), and (4.8)) that

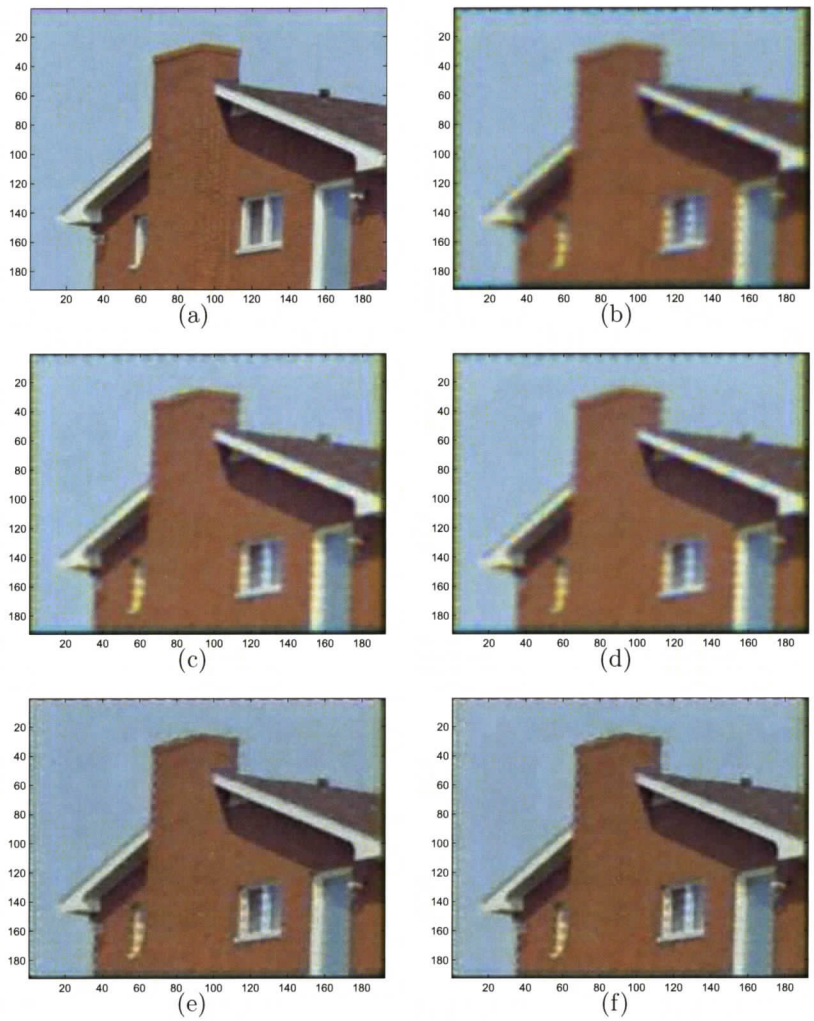
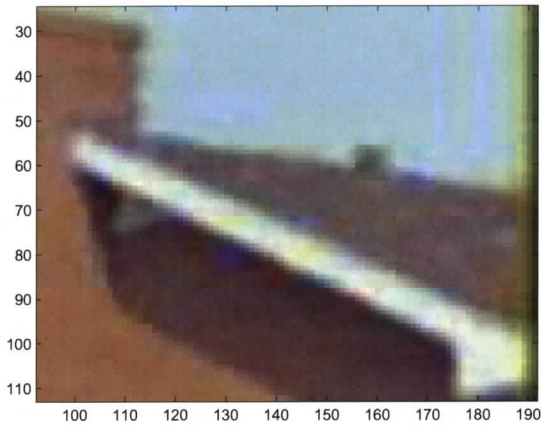
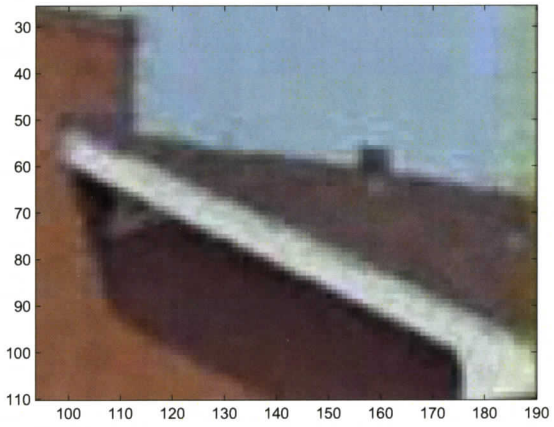


Figure 4.10: Reconstruction results and PSNR for image 'House', LR set type 'a'. (a) Original image. (b) Starting point (demosaiced shift-and-add), 19.53 dB. (c) Best SD, 20.69 dB. (d) SD+ILS, 20.03 dB. (e) DFP, 22.41 dB. (f) BFGS, 22.41 dB.

causes the CPU time for the quasi-Newton algorithms to be larger than those of their steepest-descent counterparts. The CPU time to convergence for the quasi-



(a)



(b)

Figure 4.11: Roof edge artifact: (a) for best SD reconstruction, and for (b) BFGS. The quasi-Newton reconstruction achieves a more accurate demosaicing (unnoticeable color artifacts), and sharper edges.

Newton algorithms was observed to be 10 to 20% larger than the corresponding steepest-descent mean times.

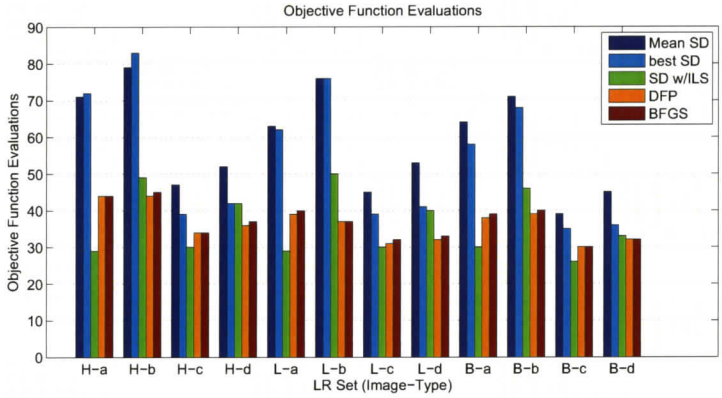


Figure 4.12: Comparative bar plot for number of objective function evaluations.

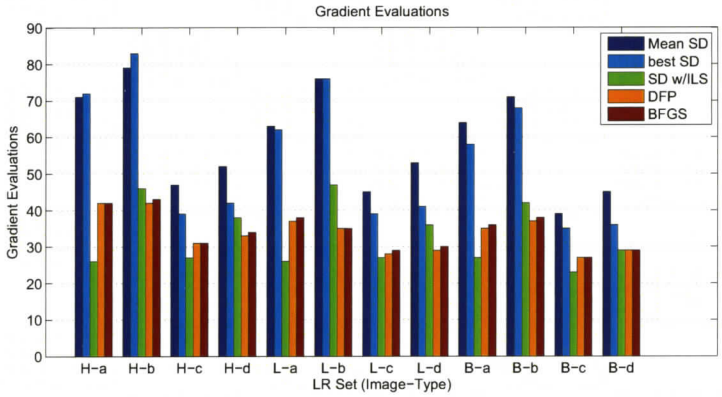


Figure 4.13: Comparative bar plot for number of gradient evaluations.

Table 4.8 summarizes the trends observed in the experiment. In this table, we show the algorithms under study sorted by time to convergence and quality of the obtained image.

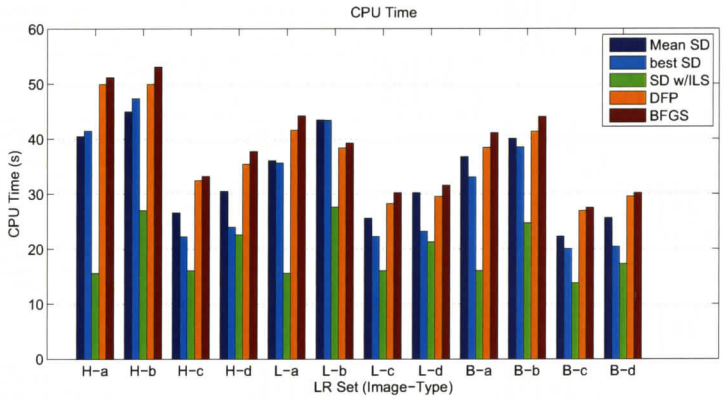


Figure 4.14: Comparative bar plot for CPU time.

Table 4.8: Trends observed in the experiment.

CPU Time (lowest to highest)	Quality (highest to lowest)
SD+ILS	BFGS
Fixed β[1]	DFP
DFP	SD+ILS
BFGS	Fixed β[1]

[1] Average performance.

In this experiment, we compared three iterative minimization methods that differ only in the way the search direction is chosen on each iteration. In the 'SD+ILS' (steepest-descent equipped with an inexact line search), the negative of the gradient is chosen as search direction. In the DFP and BFGS algorithms, the search direction is chosen in a much more sophisticated way by using (4.3) and (4.8). In all cases, the step-size is determined by the same ILS algorithm, but the quasi-Newton algorithms

achieved a much better result than algorithm 'SD+ILS'. We conclude that the search direction selection had a significant impact in the performance.

4.5 Conclusions

In this chapter, we presented the development of two sophisticated algorithms for the minimization of the costs functions proposed in [9] and [10] and reviewed in Chapter 3. By reformulating two well known quasi-Newton methods, we obtained two algorithms that are well suited to the size of the image super-resolution problem, have better convergence properties, and are less susceptible to numerical ill-conditioning. Additionally, by resorting to an inexact line search the proposed algorithms eliminate the need for selecting a step-size factor.

The proposed algorithms were used to reconstruct a series of degraded image sets and their performance was compared to that of the other approaches. Experiments were carried out for grayscale images using the objective function proposed in [9], and color images using the objective function proposed in [10]. For grayscale images, the proposed algorithms introduced a significant improvement in the computation time while obtaining images of slightly higher quality. On the other hand, for color images, the proposed algorithms obtain images of better quality at the expense of a somewhat increased computation time.

Even though the reformulation of DFP and BFGS methods lead to a significant reduction of the storage requirements when compared to their original formulation, these still require a considerable amount of storage when compared to the steepest-descent-based algorithms. In the next chapter, we design algorithms based on storage-efficient quasi-Newton methods that retain many of the desirable convergence properties of the standard BFGS method.

Chapter 5

Storage-Efficient Quasi-Newton Algorithms

5.1 Introduction

In this chapter, we propose two alternative storage-efficient quasi-Newton algorithms for the minimization of the super-resolution multi-term objective functions proposed in [9] and [10] based on the memoryless BFGS (MBFGS) algorithm [11], and the limited-memory BFGS (LBFGS) algorithm [12]. These algorithms are attractive because they retain many of the powerful convergence characteristics of the standard BFGS algorithm while requiring less storage and computational effort.

Results of the application of the proposed algorithms to sample cases indicate that, for grayscale super-resolution, the MBFGS algorithm provides the best overall improvement in terms of reconstructed image quality, storage requirements, and convergence speed when compared to a steepest-descent (SD) algorithm and other quasi-Newton algorithms. On the other hand, for color super-resolution experimental results show that the LBFGS algorithm is an effective alternative that allows one to reduce the amount of storage and computational complexity required although this

is achieved at the expense of some reduction in the quality of the images obtained.

5.2 Memoryless BFGS Algorithm

While in the standard BFGS algorithm updates are applied recursively to the approximations for the inverse Hessian, in the MBFGS algorithm, approximations of the inverse Hessian are obtained by applying the standard BFGS update formula to the identity matrix in every iteration. In this way, the BFGS updating formula

$$\boldsymbol{S_{n+1}} = \boldsymbol{S}_n + \left(1 + \frac{\boldsymbol{\gamma}_n^T \boldsymbol{S}_n \boldsymbol{\gamma}_n}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}\right) \frac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n} - \frac{\boldsymbol{\delta}_n \boldsymbol{\gamma}_n^T \boldsymbol{S}_n + \boldsymbol{S}_n \boldsymbol{\gamma}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta_n}} \tag{5.1}$$

is replaced by

$$\boldsymbol{S_{n+1}} = \boldsymbol{I} + \left(1 + \frac{\boldsymbol{\gamma}_n^T \boldsymbol{\gamma}_n}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}\right) \frac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n} - \frac{\boldsymbol{\delta}_n \boldsymbol{\gamma}_n^T + \boldsymbol{\gamma}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta_n}} \tag{5.2}$$

and the search directions are calculated as [52]

$$\begin{aligned} \boldsymbol{d}_{n+1} &= -\boldsymbol{S}_{n+1} \boldsymbol{g}_{n+1} \\ \boldsymbol{d}_{n+1} &= -\boldsymbol{g}_{n+1} + \eta_3 \boldsymbol{\gamma}_n + (\eta_2 - \eta_4) \boldsymbol{\delta}_n \end{aligned} \tag{5.3}$$

where

$$\begin{aligned} \eta_1 &= \boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n, \qquad \eta_2 = \frac{\boldsymbol{\gamma}_n^T \boldsymbol{g}_{n+1}}{\eta_1} \\ \eta_3 &= \frac{\boldsymbol{\delta_n}^T \boldsymbol{g}_{n+1}}{\eta_1}, \qquad \eta_4 = \left(1 + \frac{\boldsymbol{\gamma}_n^T \boldsymbol{\gamma}_n}{\eta_1}\right) \eta_3 \end{aligned} \tag{5.4}$$

Note that the only information needed to calculate the search direction is $\boldsymbol{g}_{n+1}$, $\boldsymbol{g}_n$, $\boldsymbol{x}_{n+1}$, and $\boldsymbol{x}_n$. The above approach can be implemented in terms of Algorithm 6.

The MBFGS algorithm was introduced by Shanno in the context of conjugate-gradient algorithms with inexact line searches [11]. In essence, Shanno derived a conjugate-gradient algorithm for which the search directions are computed as

$$\boldsymbol{d}_{n+1} = -\boldsymbol{S}_{n+1} \boldsymbol{g}_{n+1} \tag{5.5}$$

Algorithm 6: Memoryless BFGS Algorithm

$n = 0$, $\boldsymbol{x}_n = \tilde{\boldsymbol{x}}_0$, $\boldsymbol{g}_n = \nabla f(\tilde{\boldsymbol{x}}_0)$, $\boldsymbol{d}_n = -\boldsymbol{g}_n$;

until $\left|\frac{f(\boldsymbol{x}_n)-f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| < \epsilon_{THR}$ **do**

  Find an approximate solution to $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$ using an inexact line search;

  $\boldsymbol{\delta}_n = \beta_n \boldsymbol{d}_n$;

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_{n+1})$;

  $\boldsymbol{\gamma}_n = \boldsymbol{g}_{n+1} - \boldsymbol{g}_n$;

  calculate η_2, η_3, η_4 according to 5.4;

  $\boldsymbol{d}_{n+1} = -\boldsymbol{g}_{n+1} + \eta_3 \boldsymbol{\gamma}_n + (\eta_2 - \eta_4)\boldsymbol{\delta}_n$;

  store $\boldsymbol{g}_{n+1}$;

  $n = n + 1$;

$\boldsymbol{x}^* = \boldsymbol{x}_n$;

and as in the BFGS algorithm, $\boldsymbol{d}_{n+1}$ is guaranteed to be a descent direction if $\boldsymbol{\delta}_n^T \boldsymbol{\gamma}_n > 0$. In other words, if the latter inequality holds, $\boldsymbol{S}_n$ is guaranteed to be positive definite. Additionally, $\boldsymbol{S}_n$ is constructed such that the quasi-Newton equation

$$\boldsymbol{S}_{n+1} \boldsymbol{\gamma}_n = \boldsymbol{\delta}_n \tag{5.6}$$

is satisfied in every iteration. The resulting algorithm is shown to be equivalent to a BFGS quasi-Newton algorithm in which the inverse Hessian approximation is replaced by the identity matrix after each iteration. As a result, the MBFGS algorithm is essentially a conjugate-gradient algorithm that possesses improved convergence properties in ill-conditioned problems due to the positive definiteness of $\boldsymbol{S}_n$.

5.3 Limited-Memory BFGS Algorithm

The LBFGS algorithm is attractive because it retains many of the powerful convergence properties of the standard BFGS algorithm, provides control over the amount of memory required to operate, and its effectiveness has been proven for large scale optimization problems [12, 63].

As in the standard BFGS algorithm, in the LBFGS algorithm a matrix $\boldsymbol{S}_n$, which is contrived to be an approximation of the inverse Hessian, is updated in each iteration and is then multiplied by the negative gradient to obtain a new search direction. However, while in the standard BFGS algorithm the quasi-Newton matrix $\boldsymbol{S}_n$ is computed using information from all previous iterations, in the LBFGS algorithm $\boldsymbol{S}_n$ is computed using information from a limited number of past iterations, namely, the last $M - 1$ iterations. In this way a coarser but yet effective approximation of the inverse Hessian is obtained while requiring a reduced amount of storage. In addition, as we are going to show later, the LBFGS algorithm takes advantage of an efficient recursive formula to obtain the search directions.

5.3.1 Update Formula for S_n

The recursive formula for the calculation of $\boldsymbol{S}_{n+1}$, given in (5.1), can be rewritten as

$$\boldsymbol{S}_{n+1} = \boldsymbol{S}_n + S\left(\boldsymbol{\delta}_n, \boldsymbol{\gamma}_n, \boldsymbol{S}_n\right) \tag{5.7}$$

where $\boldsymbol{S}(\boldsymbol{\delta}_n, \boldsymbol{\gamma}_n, \boldsymbol{S}_n)$ is a rank-two correction matrix, which is calculated as

$$\boldsymbol{S}(\boldsymbol{\delta}_n, \boldsymbol{\gamma}_n, \boldsymbol{S}_n) = \left(1 + \frac{\boldsymbol{\gamma}_n^T \boldsymbol{S}_n \boldsymbol{\gamma}_n}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n}\right) \frac{\boldsymbol{\delta}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n} - \frac{\boldsymbol{\delta}_n \boldsymbol{\gamma}_n^T \boldsymbol{S}_n + \boldsymbol{S}_n \boldsymbol{\gamma}_n \boldsymbol{\delta}_n^T}{\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n} \tag{5.8}$$

In this way, starting from a positive definite $\boldsymbol{S}_0$, the m successive inverse Hessian approximations can be calculated as

$$\begin{aligned}
\boldsymbol{S}_1 &= \boldsymbol{S}_0 + S\left(\boldsymbol{\delta}_0, \boldsymbol{\gamma}_0, \boldsymbol{S}_0\right) \\
\boldsymbol{S}_2 &= \underbrace{\boldsymbol{S}_0 + S\left(\boldsymbol{\delta}_0, \boldsymbol{\gamma}_0, \boldsymbol{S}_0\right)}_{\boldsymbol{S}_1} + S\left(\boldsymbol{\delta}_1, \boldsymbol{\gamma}_1, \boldsymbol{S}_1\right) \\
&\vdots \\
\boldsymbol{S}_m &= \boldsymbol{S}_0 + S\left(\boldsymbol{\delta}_0, \boldsymbol{\gamma}_0, \boldsymbol{S}_0\right) + \ldots + S\left(\boldsymbol{\delta}_{m-1}, \boldsymbol{\gamma}_{m-1}, \boldsymbol{S}_{m-1}\right)
\end{aligned} \tag{5.9}$$

One effective way to reduce the storage requirements is to limit the quantity of corrections, $S(\boldsymbol{\delta}_i, \boldsymbol{\gamma}_i, \boldsymbol{S}_i)$, that are stored. If we limit the quantity of corrections to m, the storage limit would be reached after calculating $\boldsymbol{S}_m$, and to compute $\boldsymbol{S}_{m+1}$ in the next iteration we would need to discard one correction. Intuitively we would discard the oldest one, $S(\boldsymbol{\delta}_0, \boldsymbol{\gamma}_0, \boldsymbol{S}_0)$, and store the newest one, $S\left(\boldsymbol{\delta}_m, \boldsymbol{\gamma}_m, \boldsymbol{S}_m\right)$. However, it must be noticed that the updating matrices $S(\boldsymbol{\delta}_1, \boldsymbol{\gamma}_1, \boldsymbol{S}_1), \ldots, S(\boldsymbol{\delta}_{m-1}, \boldsymbol{\gamma}_{m-1}, \boldsymbol{S}_{m-1})$ depend on $S(\boldsymbol{\delta}_0 \boldsymbol{\gamma}_0, \boldsymbol{S}_0)$. So, if we simply discard this correction all others should be updated accordingly, resulting in an undesirable computation overhead.

To get around the above updating problem for the evaluation of $\boldsymbol{S}_{n+1}$, Nocedal proposed an alternative updating strategy based on the following principles [12]. First, we observe that the BFGS update formula can be written as

$$\boldsymbol{S}_{n+1} = \left(\boldsymbol{I} - \rho_n \boldsymbol{\delta}_n \boldsymbol{\gamma}_n^T\right) \boldsymbol{S}_n \left(\boldsymbol{I} - \rho_n \boldsymbol{\gamma}_n \boldsymbol{\delta}_n^T\right) + \rho_n \boldsymbol{\delta}_n \boldsymbol{\delta}_n^T \tag{5.10}$$

or

$$\boldsymbol{S}_{n+1} = \boldsymbol{V}_n^T \boldsymbol{S}_n \boldsymbol{V}_n + \rho_n \boldsymbol{\delta}_n \boldsymbol{\delta}_n^T \tag{5.11}$$

where $\rho_n = 1/\boldsymbol{\gamma}_n^T \boldsymbol{\delta}_n$. Now discarding the correction $S(\boldsymbol{\delta}_k, \boldsymbol{\gamma}_k, \boldsymbol{S}_k)$ is equivalent to letting $\boldsymbol{V}_k = \boldsymbol{I}$ and $\rho_k \boldsymbol{\delta}_k \boldsymbol{\delta}_k^T = \boldsymbol{0}$. In this way, if $m = 2$ then

$$\boldsymbol{S}_1 = \boldsymbol{V}_0^T \boldsymbol{S}_0 \boldsymbol{V}_0 + \rho_0 \boldsymbol{\delta}_0 \boldsymbol{\delta}_0^T \tag{5.12}$$

$$\boldsymbol{S}_2 = \boldsymbol{V}_1^T \boldsymbol{V}_0^T \boldsymbol{S}_0 \boldsymbol{V}_0 \boldsymbol{V}_1 + \boldsymbol{V}_1^T \rho_0 \boldsymbol{\delta}_0 \boldsymbol{\delta}_0^T \boldsymbol{V}_1 + \rho_1 \boldsymbol{\delta}_1 \boldsymbol{\delta}_1^T \tag{5.13}$$

and, after discarding the first correction ($\boldsymbol{V}_0 = \boldsymbol{I}$, $\rho_0 \boldsymbol{\delta}_0 \boldsymbol{\delta}_0^T = \boldsymbol{0}$), we get

$$\boldsymbol{S}_2 = \boldsymbol{V}_1^T \boldsymbol{S}_0 \boldsymbol{V}_1 + \rho_1 \boldsymbol{\delta}_1 \boldsymbol{\delta}_1^T \tag{5.14}$$

Proceeding in this way, the updating strategy can be generalized as follows:
For $n + 1 \leq m$,

$$\begin{aligned} \boldsymbol{S}_{n+1} = {} & \boldsymbol{V}_n^T \boldsymbol{V}_{n-1}^T \cdots \boldsymbol{V}_0^T \boldsymbol{S}_0 \boldsymbol{V}_0 \cdots \boldsymbol{V}_{n-1} \boldsymbol{V}_n \\ & + \boldsymbol{V}_n^T \cdots \boldsymbol{V}_1^T \rho_0 \boldsymbol{\delta}_0 \boldsymbol{\delta}_0^T \boldsymbol{V}_1 \cdots \boldsymbol{V}_n \\ & \vdots \\ & + \boldsymbol{V}_n^T \boldsymbol{V}_{n-1}^T \rho_{n-2} \boldsymbol{\delta}_{n-2} \boldsymbol{\delta}_{n-2}^T \boldsymbol{V}_{n-1} \boldsymbol{V}_n \\ & + \boldsymbol{V}_n^T \rho_{n-1} \boldsymbol{\delta}_{n-1} \boldsymbol{\delta}_{n-1}^T \boldsymbol{V}_n \\ & + \rho_n \boldsymbol{\delta}_n \boldsymbol{\delta}_n^T \end{aligned} \tag{5.15}$$

and, for $n + 1 > m$

$$\begin{aligned} \boldsymbol{S}_{n+1} = {} & \boldsymbol{V}_n^T \boldsymbol{V}_{n-1}^T \cdots \boldsymbol{V}_{n-m+1}^T \boldsymbol{S}_0 \boldsymbol{V}_{n-m+1} \cdots \boldsymbol{V}_{n-1} \boldsymbol{V}_n \\ & + \boldsymbol{V}_n^T \cdots \boldsymbol{V}_{n-m+2}^T \rho_{n-m+1} \boldsymbol{\delta}_{n-m+1} \boldsymbol{\delta}_{n-m+1}^T \boldsymbol{V}_{n-m+2} \cdots \boldsymbol{V}_n \\ & \vdots \\ & + \boldsymbol{V}_n^T \rho_{n-1} \boldsymbol{\delta}_{n-1} \boldsymbol{\delta}_{n-1}^T \boldsymbol{V}_n \\ & + \rho_n \boldsymbol{\delta}_n \boldsymbol{\delta}_n^T \end{aligned} \tag{5.16}$$

The proposed updating strategy entails the following properties [12]:

1. If $\boldsymbol{S}_0$ is positive definite, then $\boldsymbol{S}_n$ is positive definite provided that ${\boldsymbol{\gamma_k}}^T\boldsymbol{\delta}_k > 0$ for all $k < n$.

2. If the objective function is a strictly convex quadratic form ($f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{x}$) and the directions $\boldsymbol{\delta}_k$ are A-conjugate ($\boldsymbol{\delta}_i^T\boldsymbol{A}\boldsymbol{\delta}_j = 0$ for all $i \neq j$), then matrices $\boldsymbol{S}_n$ satisfy the quasi-Newton equation for the past m directions, i.e.,

$$\boldsymbol{S}_n\boldsymbol{\gamma}_j = \boldsymbol{\delta}_j, \quad j = n-1, \ldots, n-m \qquad (k > m) \tag{5.17}$$

3. If $m = 1$ the MBFGS update is obtained.

5.3.2 Recursive Formula for Search Directions

In the LBFGS method, as in the standard BFGS method, the line search directions are computed as

$$\boldsymbol{d}_{n+1} = -\boldsymbol{S}_{n+1}\boldsymbol{g}_{n+1} \tag{5.18}$$

In [12], a recursive formula to compute $\boldsymbol{d}_{n+1}$ is proposed based on the one developed for the standard BFGS method in [64]. This formula is convenient because it does not require storage $\boldsymbol{S}_{n+1}$ and because requires $4Qm + 2m + Q$ multiplications at most. A recursive procedure of this type can be implemented in terms of Algorithm 7. Then, the LBFGS scheme can be implemented in terms of Algorithm 8.

5.4 Results

In this section we summarize the results of experiments aimed at comparing the performance of the storage-efficient quasi-Newton algorithms to those described in Chapter 4. Five different algorithms are compared, for both, grayscale and color images, as follows:

Algorithm 7: Recursive Formula to Compute $\boldsymbol{d}_{n+1} = -\boldsymbol{S}_{n+1}\boldsymbol{g}_{n+1}$

if $n+1 \leq m$ **then**
  $I = 0$, $B = n+1$;
else
  $I = n-m+1$, $B = m$;
$\boldsymbol{q} = \boldsymbol{g}_{n+1}$;
for $i = B-1, \ldots, 0$ **do**
  $j = i + I$;
  $\alpha_i = \rho_j \boldsymbol{\delta}_j^T \boldsymbol{q}$;
  $\boldsymbol{q} = \boldsymbol{q} - \alpha_i \boldsymbol{\gamma}_i$;
$\boldsymbol{q} = \boldsymbol{S}_0 \boldsymbol{q}$;
for $i = 0, \ldots, B-1$ **do**
  $j = i + I$;
  $\beta = \rho_j \boldsymbol{\gamma}_j \boldsymbol{q}$;
  $\boldsymbol{q} = \boldsymbol{q} + \boldsymbol{\delta}_j (\alpha_i - \beta)$;
$\boldsymbol{d}_{n+1} = -\boldsymbol{q}$;

Algorithm 8: Limited-Memory BFGS Algorithm

$n = 0$, $\boldsymbol{x}_n = \tilde{\boldsymbol{x}}_0$, $\boldsymbol{g}_n = \nabla f(\tilde{\boldsymbol{x}}_0)$, $\boldsymbol{d}_n = -\boldsymbol{g}_n$, $m = M$;

until $\left|\frac{f(\boldsymbol{x}_n)-f(\boldsymbol{x}_{n-1})}{f(\boldsymbol{x}_0)}\right| < \epsilon_{THR}$ **do**

  Find an approximate solution to $\min_{\beta_n} f(\boldsymbol{x}_n + \beta_n \boldsymbol{d}_n)$ using an inexact line search;

  $\boldsymbol{\delta}_n = \beta_n \boldsymbol{d}_n$;

  $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\delta}_n$;

  $\boldsymbol{g}_{n+1} = \nabla f(\boldsymbol{x}_{n+1})$;

  $\boldsymbol{\gamma}_n = \boldsymbol{g}_{n+1} - \boldsymbol{g}_n$;

  if $n \geq m$ **then** discard $\boldsymbol{\delta}_{n-m}$, and $\boldsymbol{\gamma}_{n-m}$;

  calculate $\boldsymbol{d}_{n+1}$ using Algorithm 7;

  store $\boldsymbol{\delta}_n$, and $\boldsymbol{\gamma}_n$;

  $n = n + 1$;

$\boldsymbol{x}^* = \boldsymbol{x}_n$;

- **SD+ILS**, steepest-descent super-resolution algorithm equipped with an inexact line search.
- **BFGS**, BFGS method reformulated for super-resolution (Algorithm 3).
- **MBFGS**, memoryless BFGS reformulated for super-resolution (Algorithm 6).
- **LBFGS-5/10**, limited-memory BFGS reformulated for super-resolution (Algorithm 8) with the parameter m set to 5/10.

5.4.1 Robust Grayscale Multiframe Super-Resolution

To test the new grayscale super-resolution algorithm we constructed nine sets of degraded low-resolution images from the standard images 'Lena', 'Moon', and 'Baboon' [41], as specified in Table 5.1. Then, we constructed high-resolution images using these LR-image sets as input to the five aforementioned algorithms. In Figure

Table 5.1: LR Image-Set Specifications [1]

LR Set Type	SNR (dB)	Blur σ (pixels)
a	24	1
b	24	3
c	18	1

[1] $N = 16$, images were obtained by applying uniform translational warping, 3×3 pixels Gaussian blur, a downsample factor of 4, and additive Gaussian noise.

5.1, we give a bar plot of the number of iterations that takes each of the algorithms to converge to a solution. As expected, quasi-Newton algorithms converge much faster than the steepest-descent algorithm. Also we observe that there is no significant

difference among the different quasi-Newton algorithms. In other words, the amount of memory, i.e. the number of past iterations taken into account to estimate the inverse Hessian, does not seem to have a significant effect on the number of iterations needed to converge. This trend is also followed by the number of objective function and gradient evaluations (see Figs. 5.2 and 5.3). In Figure 5.4, we give a bar plot of

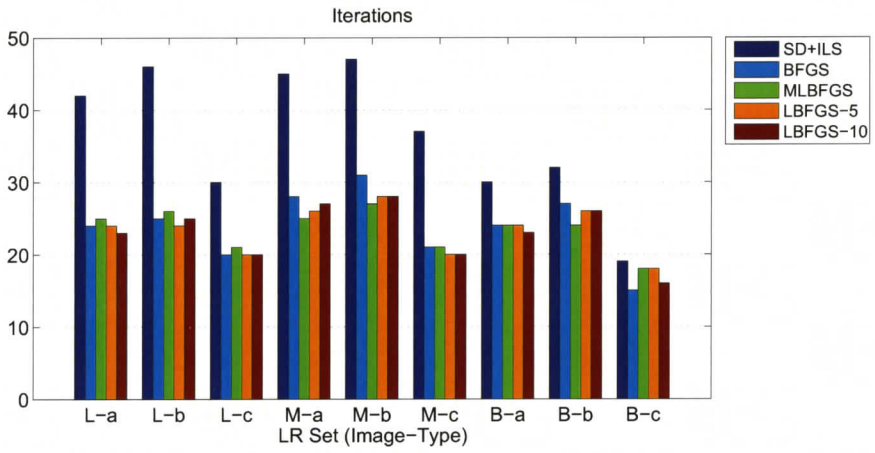


Figure 5.1: Comparative plot for the number of iterations until convergence.

the CPU time employed for the reconstruction of the different images. In this case, we observe that the BFGS algorithm takes more time to converge than its storage-efficient counterparts. This extra amount of time is due to the fact that for the standard BFGS-based algorithm the amount of calculations needed to compute the search directions increase linearly with the number of iterations. Next, in Figure 5.5, we give a bar plot of the PSNR corresponding to the HR images obtained by each algorithm. Again we observe that quasi-Newton algorithms outperform the SD+ILS algorithm for all test cases. Also, we note that there is no significant difference in the quality of the images obtained with the different quasi-Newton algorithms. In other words, the amount of memory available to the quasi-Newton-based algorithms does not have a significant effect on the quality of the images obtained.

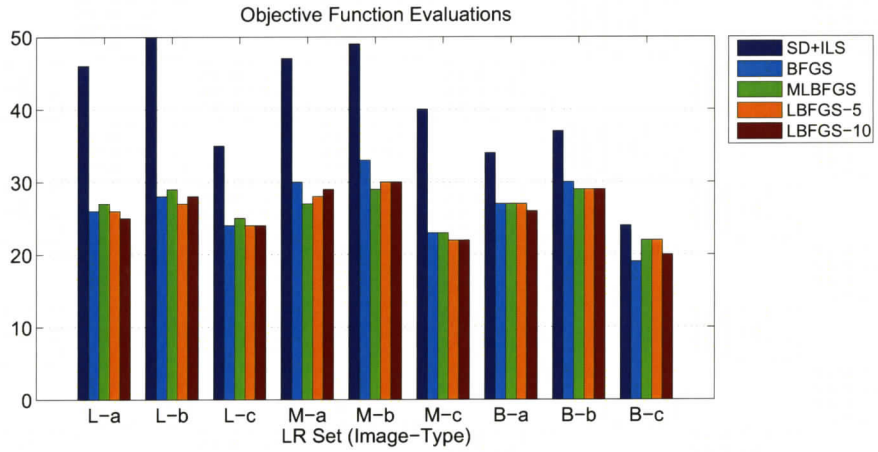


Figure 5.2: Comparative plot for the number of objective function evaluations until convergence.

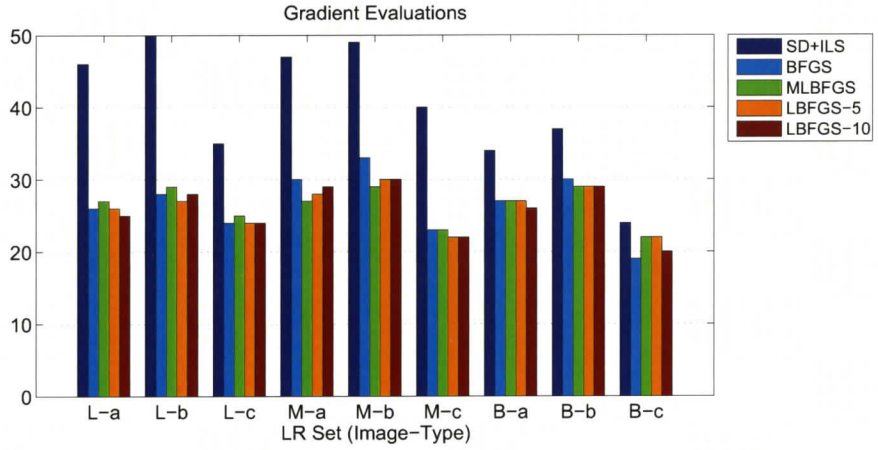


Figure 5.3: Comparative plot for the number of objective function gradient evaluations until convergence.

5.4.2 Robust Multiframe Demosaicing and Super-Resolution

To test the new color super-resolution algorithm, we constructed 12 sets of degraded low-resolution images from the standard images 'House', 'Lena', and 'Baboon' [41],

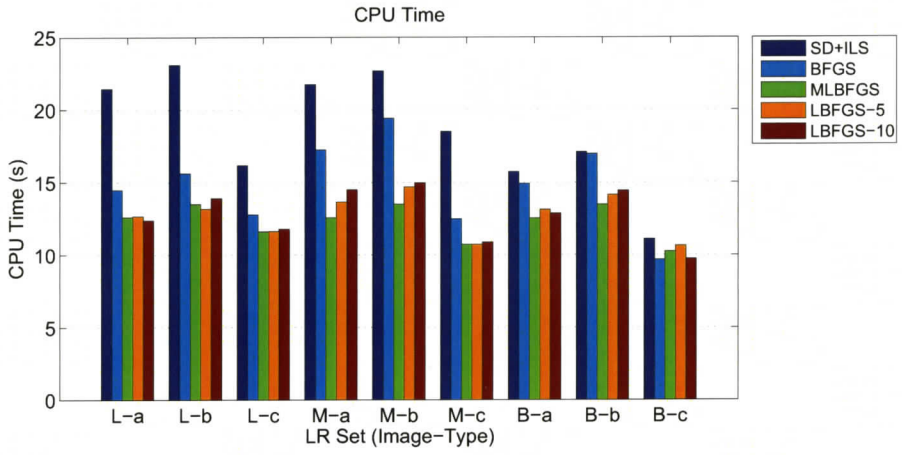


Figure 5.4: Comparative plot for CPU time for convergence.

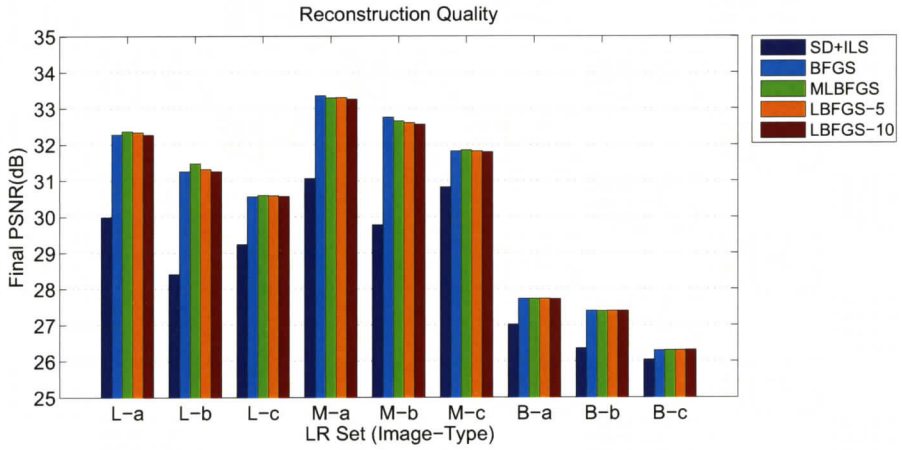


Figure 5.5: Comparative plot for reconstructed image quality given by the PSNR.

as specified in Table 5.2. Then, we constructed high-resolution images using these LR-image sets as input to the five algorithms mentioned at the beginning of this section. In Figure 5.6, we show a plot comparing the number of iterations needed for convergence. For color super-resolution quasi-Newton algorithms do not clearly outperform SD+ILS algorithms in terms of number of iterations, as was the case for

Table 5.2: LR Image-Set Specifications [1]

LR Set Type	SNR (dB)	Blur σ (pixels) [2]
a	30	1
b	30	3
c	18	1
d	18	3

[1] $N = 10$, images obtained by applying uniform translational warping, Gaussian blur, a downsample factor of 4, and additive Gaussian noise.

[2] Blur kernel size is 5×5 pixels.

grayscale super-resolution. Furthermore, no clear trend is observed when comparing the BFGS, MLBFGS, and LBFGS algorithms. For some cases the standard BFGS performs better than the other quasi-Newton algorithms, e.g., for set 'H-b', and in other cases it performs worst, like for set 'H-a'. The same behaviour is observed for the number of objective function and gradient evaluations (Figs. 5.7, and 5.8).

When observing the results obtained for the CPU-time comparison we note that the standard BFGS algorithm running time is well above those corresponding to the other algorithms. Again, we attribute this behaviour to the increased computational complexity associated with the calculation of the search directions for the BFGS algorithm. We also observe that, in general, increasing the memory of the quasi-Newton algorithms, by increasing parameter M, results in an increase of the computation time. This is expected as the calculations of the search directions get more complex as more iterations are required (see Fig. 5.10).

In Figure 5.11, we give a bar plot for the PSNR corresponding to the HR images

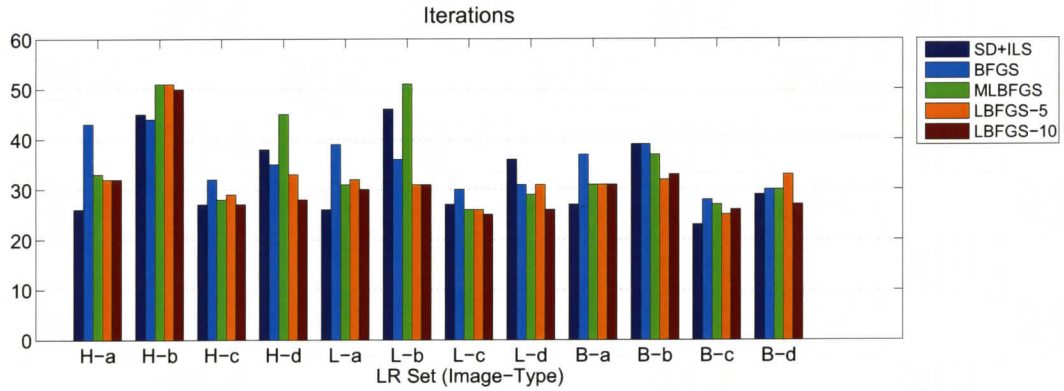


Figure 5.6: Comparative plot for number of iterations until convergence.

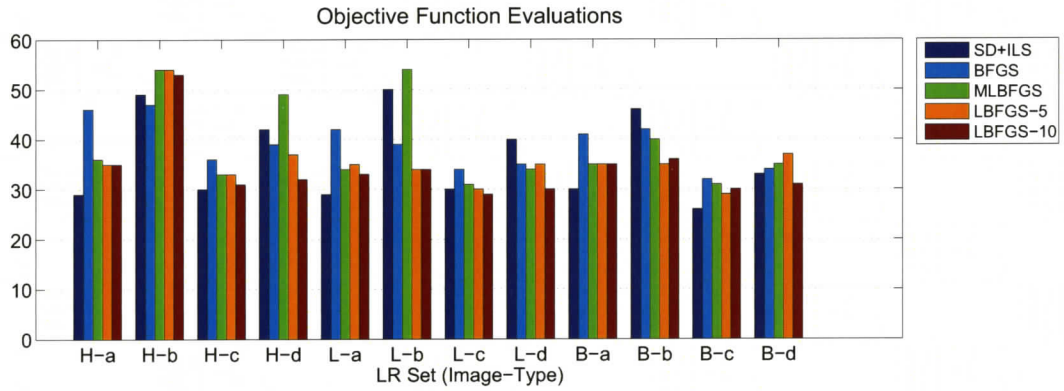


Figure 5.7: Comparative plot for number of objective function evaluations until convergence.

obtained with the various algorithms. Clearly, quasi-Newton algorithms outperform the SD+ILS algorithm. In particular, the standard BFGS algorithm performs best of all and, in general, we observe that, for the limited-memory algorithms, the quality of the obtained images improves as more storage resources are made available (see Fig. 5.12).

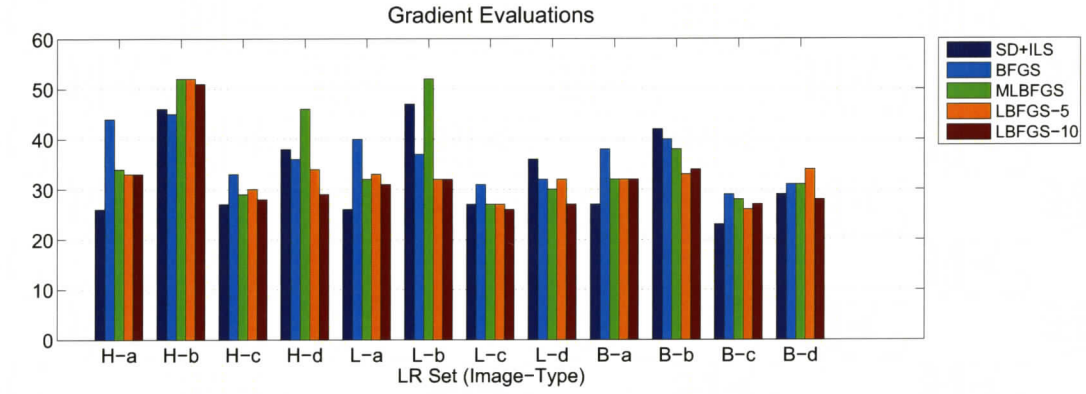


Figure 5.8: Comparative plot for number of objective function gradient evaluations until convergence.

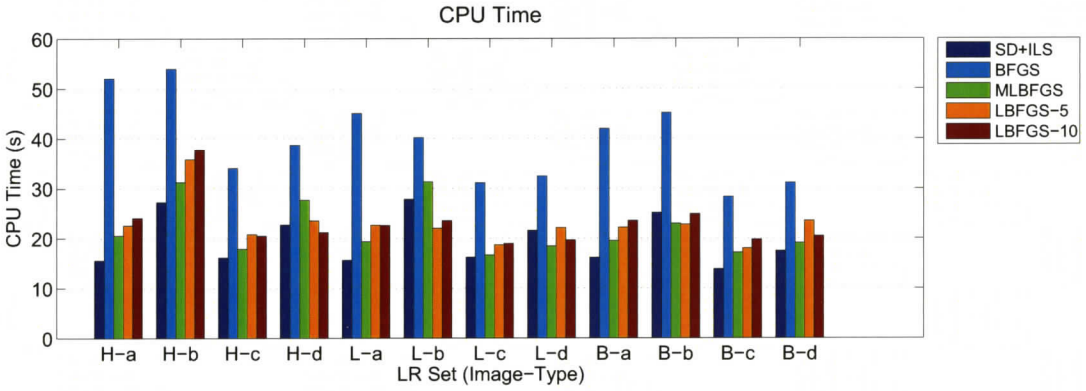


Figure 5.9: Comparative plot for CPU time for convergence.

5.5 Conclusions

We applied two storage-efficient alternatives to the quasi-Newton SR algorithms proposed in Chapter 4, namely, the memoryless and limited-memory BFGS algorithms. The two algorithms were shown to yield images of a quality comparable to those obtained using the standard BFGS algorithm and, at the same time, require a significantly reduced amount of memory storage and computation effort.

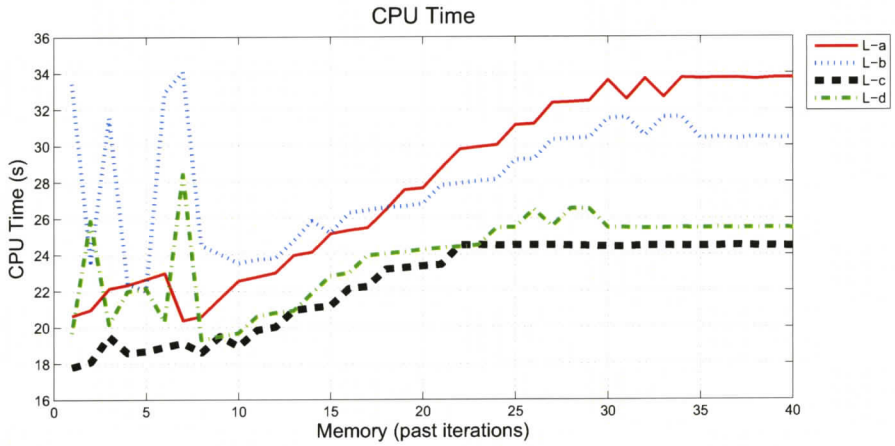


Figure 5.10: LBFGS algorithm — CPU time for convergence for different memory settings (image: 'Lena').

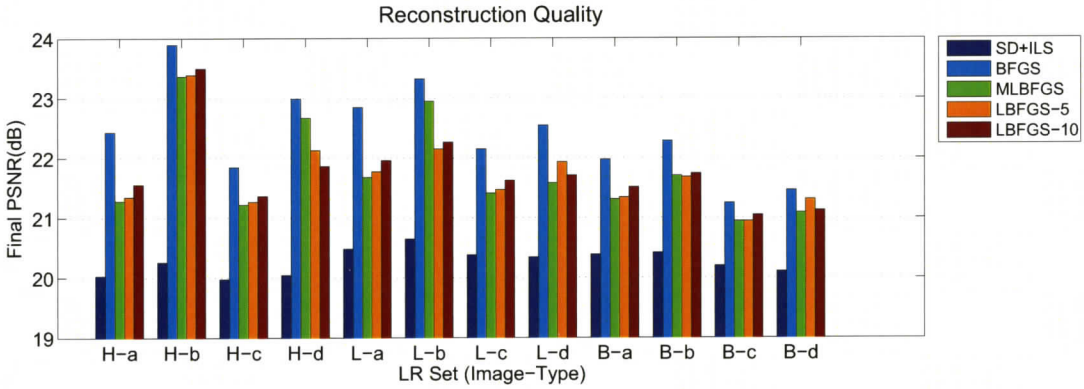


Figure 5.11: Comparative plot for reconstructed image quality given by the PSNR.

When compared to the steepest-descent algorithms, the memoryless BFGS algorithm introduces a significant improvement in terms of the quality of the obtained images at the expense of some extra storage and computational effort.

For the case of grayscale SR, we observed negligible differences in the quality of the images obtained with the different quasi-Newton algorithms. We conclude,

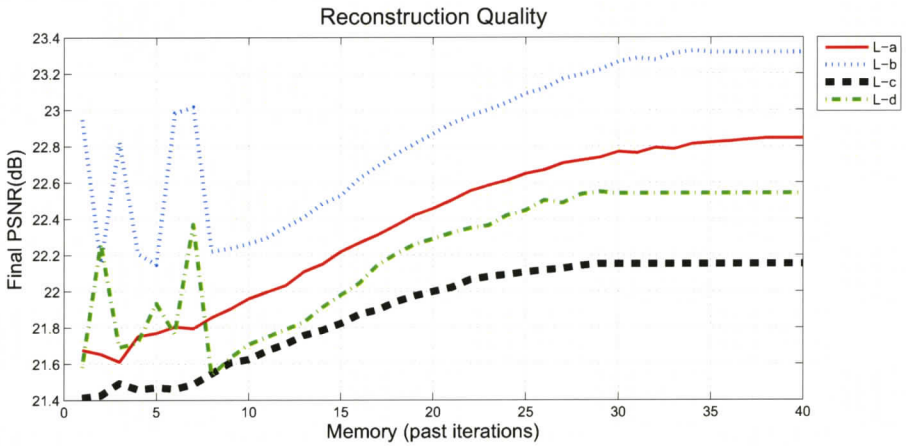


Figure 5.12: LBFGS algorithm — HR image PSNR for different memory settings (image: 'Lena').

therefore, that given its low demand for storage and computational effort, the MBFGS algorithm is an excellent choice for minimizing the grayscale super-resolution objective function proposed in [9].

For hybrid demosaicing and color SR, we observed that the quality of the images obtained, as measured by the PSNR, improves when additional memory is made available to the quasi-Newton algorithms. However, this benefit comes at the expense of increased computational effort on top of the additional storage requirements. We conclude, therefore, that when minimizing the color super-resolution objective-function proposed in [10], the LBFGS algorithm is a very good alternative as it leads to a meaningful compromise between the amount of storage and computational effort required, on the one hand, and the quality of the images obtained on the other.

Chapter 6

Conclusions and Future Work

The main objective of this thesis has been to obtain efficient super-resolution algorithms through the introduction of key improvements to two state-of-the-art techniques developed in [9] and [10]. These improvements are based on the adaptation and implementation of powerful quasi-Newton minimization algorithms. Several alternatives have been proposed and experimental results show that a significant improvement in terms of convergence speed and image quality can be achieved. Next, the contributions and conclusions of this thesis are summarized and suggestions for further research are presented.

6.1 Conclusions

In Chapter 2, we presented an overview of the most representative super-resolution techniques. These techniques have been grouped according to their domain of formulation into those based on the frequency domain, those based on the spatial domain, and those based on the wavelet domain. We showed that frequency-domain-based methods are of simple formulation and computationally efficient but can only deal with translational motion models and space-independent degradations. Spatial-

domain methods, on the other hand, are computationally more expensive but can work with more complex motion models and degradations and can deal with spatially dependent regularizing functionals that improve the numerical conditioning of their associated optimization problem. Towards the end of the chapter, we presented some wavelet-based algorithms which can produce an improved resolution image from a single image without requiring several sub-pixel displaced images.

In Chapter 3, we described two state-of-the-art super-resolution algorithms and gave details about their implementation. Then we pointed out two fundamental aspects that could be improved. First, these algorithms rely on a minimization algorithm that is quite innefficient given the ill-conditioned nature of the super-resolution problem. Second, a fundamental tuning parameter is left to be provided by the user and, consequently, the algorithm's effectiveness is dependant on the user's ability to select a value for this parameter.

In Chapter 4, we proposed two sophisticated algorithms for the minimization of the objective functions proposed in [9] and [10] and reviewed in Chapter 3. By reformulating the DFP and BFGS methods, we obtained two algorithms that are well suited to the size of the image super-resolution problem, have better convergence properties, and are less susceptible to numerical ill-conditioning. Additionally, by resorting to an inexact line search, the proposed algorithms eliminate the need for user-defined parameters.

The proposed algorithms were used to reconstruct a series of degraded image sets and their performance was compared to that of the algorithms proposed in [9] and [10]. It was shown that for grayscale images the proposed algorithms introduced a significant improvement in the computation time while obtaining images of slightly better quality. For color images, on the other hand, the proposed algorithms yield images of better quality at the expense of a somewhat increased computation time.

In Chapter 5, we presented two memory efficient alternatives to the algorithms

proposed in Chapter 4. These algorithms are based on the memoryless BFGS and limited-memory BFGS methods and were shown to achieve images of a quality comparable to those obtained with the BFGS algorithm of Chapter 4 and at the same time require a significantly less storage and computation effort. As was shown, when compared to the steepest-descent-based algorithms, the memoryless BFGS algorithm introduces a significant improvement in terms of the quality of the obtained images at the expense of little extra storage and computational complexity.

Experimental results indicate that given its low storage demands and computational complexity the MBFGS-based algorithm should be the method of choice when minimizing the grayscale super-resolution objective function proposed in [9]. On the other hand, for color super-resolution we observed that the quality of the images obtained as measured by the PSNR improves when additional memory is available to the quasi-Newton algorithms. Therefore, we conclude that when minimizing the color super-resolution objective function proposed in [10], the LBFGS algorithm constitutes an effective alternative as it leads to a meaningful compromise between the amount of storage and computational effort required, on the one hand, and the quality of the images obtained on the other.

6.2 Future Work

Some directions for future research are as follows:

- *Stopping criterion refinement:* Currently, the stopping criterion is designed to halt iterations when no significant reduction of the objective function value is being made. The rationale behind such criterion is to avoid computations that will not provide a significant improvement in the quality of the reconstruction. However, one important point to note is that a reduction in the objective function value does not imply an increase in the PSNR. In effect, it has been

observed that the PSNR increases at the beggining of the iterations and peaks before a minimum of the objective function is reached. That is, it can happen that after a certain number of iterations the PSNR starts decreasing even though the objective function is decreasing. An interesting research direction would be the development of an optimal stopping criterion that can capture the PSNR peak thereby eliminating iterations that would degrade the quality of the image.

- *Inexact line search tuning:* The introduction of an inexact line search in the iterative process eliminated the need for selecting parameter β. However, as can be seen from line 1 of Algorithm 4, the inexact line search itself has several parameters that can be tuned. For our implementation, we used a set of values that has been shown to work well for a variety of problems [52] and found that the same set of values give good results in SR. One possible direction of research would be to establish a clear understanding on how the values of these parameters can be optimally selected for the SR problem. Tuning these parameters to the SR problem could bring about improvements in terms of convergence time and/or the quality of the images obtained.

- λ *selection:* As detailed in Chapter 3, grayscale SR relies on the minimization of a two-term objective function, i.e.,

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left[J(\boldsymbol{x}) + \lambda \rho(\boldsymbol{x}) \right] \tag{6.1}$$

where a data-fidelity term, $J(\boldsymbol{x})$, assures a certain fidelity or closeness of the final solution to the observed data; a regularization term, $\rho(\boldsymbol{x})$, improves the numerical conditioning of the problem; and a constant, λ, known as *regularization parameter*, applies relative weighting to the second term.

Throughout this work, the value of λ was empirically adjusted to obtain good quality reconstructions. In fact, the final PSNR value was observed to be strongly dependant on the appropriate selection of λ.

There are numerical methods for the appropriate and automatic selection of the regularization parameter, like the *'L-curve'* method of [65] or the *'generalized cross-validation'* method of [66]. However, these are designed for L_2-norm objective functions. In effect, an interesting direction of research would be to investigate if such methods could be adapted to robust objective functions based on the L_1 norm.

Bibliography

[1] K. Parulski and M. Rabbani, "The continuing evolution of digital cameras and digital photography systems," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, May 2000, pp. 101–104.

[2] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, May 2003.

[3] T. Komatsu, K. Aizawa, T. Igarashi, and T. Saito, "Signal-processing based method for acquiring very high resolution images with multiple cameras and its theoretical analysis," *IEE Communications, Speech and Vision*, vol. 140, no. 1, pp. 19–24, 1993.

[4] N. Nguyen, *Numerical Algorithms for Image Superresolution.* Standford University, 2000.

[5] H. C. Andrews and B. Hunt, *Digital Image Restoration*, A. V. Oppenheim, Ed. Prentice hall, 1977.

[6] R. Tsai and T. Huang, "Multiframe image restoration and registration," *Advances in Computer Vision and Image Processing*, vol. 1, pp. 317–339, 1984.

[7] S. Borman and R. Stevenson, *Spatial Resolution Enhancement of Low-Resolution Image Sequences. A Comprehensive Review with Directions for Future Research.* University of Notre Dame, 1998.

[8] C. Segall, R. Molina, A. Katsaggelos, and J. Mateos, "Reconstruction of high-resolution image frames from a sequence of low-resolution and compressed observations," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2002, pp. 1701–1704.

[9] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Transactions on Image Processing*, vol. 13, pp. 1327–1343, 2004.

[10] S. Farsiu, M. Elad, and P. Milanfar, "Multiframe demosaicing and super-resolution of color images," *IEEE Transactions on Image Processing*, vol. 15, pp. 141–159, 2006.

[11] D. F. Shanno, "Conjugate gradient methods with inexact searches," *Mathematics of Operations Research*, vol. 3, pp. 244–256, August 1978.

[12] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of Computation*, vol. 35, pp. 773–782, 1980.

[13] A. Antoniou, *Digital Signal Processing: Signal, Systems, and Applications.* McGraw-Hill, 2005.

[14] A. Tekalp, M. Ozkan, and M. Sezan, "High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, March 1992, pp. 169–172.

[15] H. Wei and T. Binnie, "High-resolution image reconstruction from multiple low-resolution images," in *Proc. Seventh International Conference on Image Processing and Its Applications*, vol. 2, 1999, pp. 596–600.

[16] S. Kim, N. Bose, and H. Valenzuela, "Recursive reconstruction of high resolution image from noisy undersampled multiframes," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 6, pp. 1013–1027, 1990.

[17] S. Kim and W.-Y. Su, "Recursive high-resolution reconstruction of blurred multiframe images," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 534–539, 1993.

[18] G. H. Golub and C. F. Van-Loan, *Matrix Computations.* Baltimore: The John Hopkins University Press, 1989.

[19] N. Bose, H. Kim, and H. Valenzuela, "Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, April 1993, pp. 269–272.

[20] C. Davila, "Recursive total least squares algorithms for adaptive filtering," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.*, vol. 3, 14-17 April 1991, pp. 1853–1856.

[21] J. Brown, J., "Multi-channel sampling of low-pass signals," *IEEE Transactions on Circuits and Systems*, vol. 28, no. 2, pp. 101–106, 1981.

[22] A. Papoulis, "Generalized sampling expansion," *IEEE Transactions on Circuits and Systems*, vol. 24, no. 11, pp. 652–654, 1977.

[23] H. Ur and D. Gross, "Improved resolution from subpixel shifted pictures," *CVGIP: Graph. Models Image Process.*, vol. 54, no. 2, pp. 181–186, 1992.

[24] H. Shekarforoush and R. Chellappa, "Data-driven multichannel superresolution with application to video sequences," *Journal of the Optical Society of America A-Optics Image Science and Vision*, vol. 16, no. 3, pp. 481–492, Mar. 1999.

[25] D. Keren, S. Peleg, and R. Brada, "Image sequence enhancement using sub-pixel displacements," in *Proc. Computer Society Conference on Computer Vision and Pattern Recognition*, June 1988, pp. 742–746.

[26] K. Aizawa, T. Komatsu, and T. Saito, "Acquisition of very high resolution images using stereo cameras," *Visual Communications and Image Processing '91: Visual Communication*, vol. 1605, no. 1, pp. 318–328, 1991.

[27] J. Clark, M. Palmer, and P. Lawrence, "A transformation method for the reconstruction of functions from nonuniformly spaced samples," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 5, pp. 1151–1165, 1985.

[28] N. Nguyen and P. Milanfar, "A wavelet-based interpolation-restoration method for superresolution (wavelet superresolution)," *Circuit Systems Signal Processing*, vol. 19, pp. 321–338, 2000.

[29] C. Ford and D. Etter, "Wavelet-based interpolation method for nonuniformly sampled signals," *Wavelet applications in signal and image processing IV*, vol. 1, pp. 14–23, 1996.

[30] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP: Graph. Models Image Process.*, vol. 53, no. 3, pp. 231–239, 1991.

[31] K. S. Shanmugan and A. Breipohl, *Random Signals. Detection, Estimation and Data Analysis.* Wiley, 1988.

[32] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, pp. 1646–1658, 1997.

[33] R. Schultz, R. Schultz, and R. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Transactions on Image Processing*, vol. 5, no. 6, pp. 996–1011, 1996.

[34] B. Tom and A. Katsaggelos, "Reconstruction of a high-resolution image by simultaneous registration, restoration, and interpolation of low-resolution images," in *Proc. International Conference on Image Processing*, vol. 2, Oct. 1995, pp. 539–542.

[35] R. Hardie, K. Barnard, and E. Armstrong, "Joint map registration and high-resolution image estimation using a sequence of undersampled images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1621–1633, 1997.

[36] A. Temizel, A. Temizel, and T. Vlachos, "Wavelet domain image resolution enhancement," *IEE Proceedings Vision, Image and Signal Processing*, vol. 153, no. 1, pp. 25–30, 2006.

[37] S. Chang, S. Chang, Z. Cvetkovic, and M. Vetterli, "Resolution enhancement of images using wavelet transform extrema extrapolation," in *International Conference on Acoustics, Speech, and Signal Processing*, Z. Cvetkovic, Ed., vol. 4, 1995, pp. 2379–2382.

[38] M. Elad and Y. Hel-Or, "A fast super-resolution reconstruction algorithm for pure translational motion and common space-invariant blur," *IEEE Transactions on Image Processing*, vol. 10, pp. 1187–1193, 2001.

[39] N. Nguyen, P. Milanfar, and G. Golub, "A computationally efficient image superresolution algorithm," *IEEE Transactions on Image Processing*, vol. 10, pp. 573–583, 2001.

[40] M. Elad and A. Feuer, "Superresolution restoration of an image sequence: adaptive filtering approach," *IEEE Transactions on Image Processing*, vol. 8, pp. 387–395, 1999.

[41] University of Southern California. Signal and Image Processing Institute (SIPI) image database. [Online]. Available: http://sipi.usc.edu/database/. Last accessed: August 2007.

[42] P. Huber, *Robust Statistics*. New York: Wiley, 1981.

[43] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[44] N. Shah and A. Zakhor, "Resolution enhancement of color video sequences," *IEEE Transactions on Image Processing*, vol. 8, no. 6, pp. 879–885, 1999.

[45] B. Tom and A. Katsaggelos, "Resolution enhancement of monochrome and color video using motion compensation," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 278–287, 2001.

[46] B. E. Bayer, "United states patent 3,971,065: Color imaging array," July 1976.

[47] D. Keren and M. Osadchy, "Restoring subsampled color images," *Vis. Appl.*, vol. 11, pp. 197–202, 1999.

[48] R. Kimmel, "Demosaicing: image reconstruction from color ccd samples," *IEEE Transactions on Image Processing*, vol. 8, no. 9, pp. 1221–1228, Sept. 1999.

[49] J. C. Russ, *The Image Processing Handbook.* IEEE Press, 1995.

[50] M. K. Ng and N. K. Bose, "Mathematical analysis of super-resolution methodology," *IEEE Signal Processing Magazine*, vol. 20 Iss. 3, pp. 62–74, 2003.

[51] Z. Wang and F. Qi, "Analysis of multiframe super-resolution reconstruction for image anti-aliasing and deblurring," *Image and Vision Computing*, vol. 23, pp. 393–404, 2005.

[52] A. Antoniou and W. S. Lu, *Practical Optimization: Algorithms and Engineering Applications.* New York: Springer, 2007.

[53] D. S. Watkins, *Fundamentals of Matrix Computations.* New York: Wiley, 2002.

[54] W. Davidon, "Variable metric method for minimization," Argonne National Lab., Lemont, Ill., Tech. Rep. ANL-5990, 1959.

[55] R. Fletcher and M. Powell, "A rapidly convergent descent method for minimization," *The Computer Journal*, vol. 6, pp. 163–168, 1963.

[56] C. Broyden, "Quasi-Newton methods and their application to function minimisation," *Mathematics of Computation*, vol. 21, pp. 368–381, 1967.

[57] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, pp. 317–322, 1970.

[58] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Mathematics of Computation*, vol. 24, no. 109, pp. 23–26, Jan. 1970.

[59] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, July 1970.

[60] R. Fletcher, *Practical Methods of Optimization.* New York: John Wiley & Sons, 1980, vol. 1.

[61] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau*, vol. 49, pp. 409–436, 1952.

[62] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.

[63] D. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[64] H. Matthies and G. Strang, "The solution of nonlinear finite elements equations," *International Journal for Numerical Methods in Engineering*, vol. 14, pp. 1613–1626, 1979.

[65] P. C. Hansen, "The L-curve and its use in the numerical treatment of inverse problems," in *Computational Inverse Problems in Electrocardiology*, ser. Advances in Computational Bioengineering, P. Johnston, Ed. WIT Press, 2000, pp. 119–142.

[66] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Transactions on Image Processing*, vol. 10, no. 9, pp. 1299–1308, 2001.

Appendix A

Mathematical Proofs

A.1 ML estimator in the presence of additive white Gaussian noise

In the following, we show that

$$\underset{\boldsymbol{x}}{\text{Maximize}}\, P(\{\boldsymbol{y_1},\ldots,\boldsymbol{y_N}\}|\boldsymbol{x}) \equiv \underset{\boldsymbol{x}}{\text{Minimize}} \left[\sum_{k=1}^{N} \|\boldsymbol{DF_kHx} - \boldsymbol{y_k}\|_2^2\right] \tag{A.1}$$

where $P(\{\boldsymbol{y_1},\ldots,\boldsymbol{y_N}\}|\boldsymbol{x})$ is the likelihood function (as defined in [31]), and

$$\boldsymbol{y_k} = \boldsymbol{DF_kHx} + \boldsymbol{\nu_k} \tag{A.2}$$

where $\boldsymbol{\nu_k}$ has a multivariate Gaussian distribution thereby representing additive white Gaussian noise.

Considering A.2, the probability of acquiring an LR image $\boldsymbol{y_k}$, $P(\boldsymbol{y_k}|\boldsymbol{x})$, is equal to the probability of the noise being $\boldsymbol{\nu_k} = \boldsymbol{y_k} - \boldsymbol{DF_kHx}$. Then, since we assume $\boldsymbol{\nu_k}$ to have a multivariate Gaussian distribution,

$$\begin{aligned} P(\boldsymbol{y_k}|\boldsymbol{x}) &= P(\boldsymbol{\nu_k})|_{\boldsymbol{\nu_k}=\boldsymbol{y_k}-\boldsymbol{DF_kHx}} \\ &= \underbrace{\frac{1}{(2\pi)^{M^2} det(C)^{1/2}}}_{\alpha} e^{-1/2(\boldsymbol{y_k}-\boldsymbol{DF_kHx})^T \boldsymbol{C}^{-1}(\boldsymbol{y_k}-\boldsymbol{DF_kHx})} \end{aligned} \tag{A.3}$$

Now, considering the acquisition of each LR image as an statistically independent event, the probability of acquiring an LR images set $\{\boldsymbol{y_1}, \ldots, \boldsymbol{y_N}\}$ is equal to the product of the probabilities of acquiring each individual LR image, that is

$$\begin{aligned} P(\{\boldsymbol{y_1}, \ldots, \boldsymbol{y_N}\}|\boldsymbol{x}) &= \prod_{k=1}^{N} P(\boldsymbol{y_k}|\boldsymbol{x}) \\ &= \alpha^N e^{-1/2 \sum_{k=1}^{N} (\boldsymbol{y_k} - \boldsymbol{DF_k Hx})^T \boldsymbol{C}^{-1} (\boldsymbol{y_k} - \boldsymbol{DF_k Hx})} \end{aligned} \tag{A.4}$$

Since the logarithm is a monotonic increasing function we can write

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{Maximize}} \left[P(\{\boldsymbol{y_1}, \ldots, \boldsymbol{y_N}\}|\boldsymbol{x})\right] \\ &\equiv \underset{\boldsymbol{x}}{\text{Maximize}} \{ln \left[P(\{\boldsymbol{y_1}, \ldots, \boldsymbol{y_N}\}|\boldsymbol{x})\right]\} \\ &\equiv \underset{\boldsymbol{x}}{\text{Maximize}} \left[N ln(\alpha) - \frac{1}{2} \sum_{k=1}^{N} (\boldsymbol{y_k} - \boldsymbol{DF_k Hx})^T \boldsymbol{C}^{-1} (\boldsymbol{y_k} - \boldsymbol{DF_k Hx}) \right] \\ &\equiv \underset{\boldsymbol{x}}{\text{Minimize}} \left[\sum_{k=1}^{N} (\boldsymbol{y_k} - \boldsymbol{DF_k Hx})^T \boldsymbol{C}^{-1} (\boldsymbol{y_k} - \boldsymbol{DF_k Hx}) \right] \end{aligned} \tag{A.5}$$

If we assume white noise, the inverse of the covariance matrix is $\boldsymbol{C}^{-1} = \sigma^{-2}\boldsymbol{I}$ and $\hat{\boldsymbol{x}}$ can be found by solving the problem

$$\underset{\boldsymbol{x}}{\text{Minimize}} \left[\sum_{k=1}^{N} \|\boldsymbol{DF_k Hx} - \boldsymbol{y_k}\|_2^2 \right] \tag{A.6}$$

A.2 Derivation of objective function gradients

A.2.1 Grayscale objective function

In the following, we show that

$$\begin{aligned} \boldsymbol{g_p} &= \frac{d}{d\boldsymbol{z}} \left(\sum_{k=1}^{N} \|\boldsymbol{DF_k z} - \boldsymbol{y_k}\|_p^p \right) \\ &= p \sum_{k=1}^{N} \boldsymbol{F_k^T D^T} \text{sign}(\boldsymbol{DF_k z} - \boldsymbol{y_k}) \odot |\boldsymbol{DF_k z} - \boldsymbol{y_k}|^{p-1} \end{aligned} \tag{A.7}$$

where $\odot$ is the element-by-element multiplication operator.

We start our derivation by observing that if $\boldsymbol{x} \in \mathbb{R}^M$, then

$$\begin{aligned} \nabla \|\boldsymbol{x}\|_p^p &= \nabla(|x_1|^p + \cdots + |x_M|^p) \\ &= \left[\text{sign}(x_1)p|x_1|^{p-1}, \ldots, \text{sign}(x_M)p|x_M|^{p-1}\right] \\ &= \text{sign}(\boldsymbol{x}) \odot p|\boldsymbol{x}|^{p-1} \end{aligned} \tag{A.8}$$

Considering the gradient identity

$$\left[\frac{d}{d\boldsymbol{x}} U(\boldsymbol{w})\right]_i = \left\langle \nabla U(\boldsymbol{w}), \frac{d}{dx_i}\boldsymbol{w}(\boldsymbol{x}) \right\rangle \tag{A.9}$$

and letting $U(\boldsymbol{w}) = \|\boldsymbol{w}\|_p^p$ where $\boldsymbol{w} = \boldsymbol{A}\boldsymbol{x}$ and $\boldsymbol{A}$ is an $M \times M$ matrix, we have

$$\left(\frac{d}{d\boldsymbol{x}}\|\boldsymbol{A}\boldsymbol{x}\|_p^p\right)_i = \left\langle \text{sign}(\boldsymbol{A}\boldsymbol{x}) \odot p|\boldsymbol{A}\boldsymbol{x}|^{p-1}, \frac{d}{dx_i}\boldsymbol{A}\boldsymbol{x} \right\rangle \tag{A.10}$$

It is easy to see that $d(\boldsymbol{A}\boldsymbol{x})/dx_i = \boldsymbol{c_i}$ where $\boldsymbol{c_i}$ is the i^{th} column of $\boldsymbol{A}$; therefore,

$$\frac{d}{d\boldsymbol{x}}\|\boldsymbol{A}\boldsymbol{x}\|_p^p = \boldsymbol{A^T}\text{sign}(\boldsymbol{A}\boldsymbol{x}) \odot p|\boldsymbol{A}\boldsymbol{x}|^{p-1} \tag{A.11}$$

By letting $\boldsymbol{A} = \boldsymbol{D}\boldsymbol{F_k}$, $\boldsymbol{x} = \boldsymbol{z} - \boldsymbol{F_k}^{-1}\boldsymbol{D^T}\boldsymbol{y_k}$, and noting that $d\boldsymbol{x} = d\boldsymbol{z}$, we have

$$\frac{d}{d\boldsymbol{z}}\|\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}\|_p^p = p\boldsymbol{F_k^T}\boldsymbol{D^T}\text{sign}(\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}) \odot |\boldsymbol{D}\boldsymbol{F_k}\boldsymbol{z} - \boldsymbol{y_k}|^{p-1} \tag{A.12}$$

Finally, using the linearity property of the gradient and (A.12) we obtain (A.7).

A.2.2 Color objective function

First, we notice that the multi-term objective function corresponding to the multi-frame demosaicing and color super-resolution problem and its gradient are:

$$J(\boldsymbol{x}) = J_0(\boldsymbol{x}) + \lambda' J_1(\boldsymbol{x}) + \lambda'' J_2(\boldsymbol{x}) + \lambda''' J_3(\boldsymbol{x}) \tag{A.13}$$

$$\nabla J(\boldsymbol{x}) = \nabla J_0(\boldsymbol{x}) + \lambda' \nabla J_1(\boldsymbol{x}) + \lambda'' \nabla J_2(\boldsymbol{x}) + \lambda''' \nabla J_3(\boldsymbol{x}) \tag{A.14}$$

where

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_R} \\ \boldsymbol{x_G} \\ \boldsymbol{x_B} \end{pmatrix} \tag{A.15}$$

and the gradient can be expressed as

$$\nabla J_q(\boldsymbol{x}) = \begin{pmatrix} \nabla_{\boldsymbol{x_R}} J_q \\ \nabla_{\boldsymbol{x_G}} J_q \\ \nabla_{\boldsymbol{x_B}} J_q \end{pmatrix} \tag{A.16}$$

Under these circumstances, the gradients for each objective function term can be derived as follows:

1. *Data Fidelity Term:*

$$\nabla_{\boldsymbol{x_i}} J_0 = \nabla_{\boldsymbol{x_i}} \| \boldsymbol{\Phi_i} (\boldsymbol{H x_i} - \hat{\boldsymbol{z}}_{\boldsymbol{i}}) \|_1 \tag{A.17}$$

$$= \nabla_{\boldsymbol{x_i}} \| \underbrace{\boldsymbol{\Phi_i H}}_{\boldsymbol{A}} \underbrace{\left(\boldsymbol{x_i} - \boldsymbol{H^{-1}} \hat{\boldsymbol{z}}_{\boldsymbol{i}}\right)}_{\boldsymbol{x}} \|_1 \tag{A.18}$$

Noting that $d\boldsymbol{x_i} = d\boldsymbol{x}$, (A.11) gives

$$\nabla_{\boldsymbol{x_i}} J_0 = (\boldsymbol{\Phi_i H})^T \operatorname{sign} (\boldsymbol{\Phi_i H x_i} - \boldsymbol{\Phi_i} \hat{\boldsymbol{z}}_{\boldsymbol{i}}) \tag{A.19}$$

This expression can be further simplified as

$$\nabla_{\boldsymbol{x_i}} J_0 = \boldsymbol{H}^T \boldsymbol{\Phi_i} \operatorname{sign} (\boldsymbol{H x_i} - \hat{\boldsymbol{z}}_{\boldsymbol{i}}) \tag{A.20}$$

by noticing that $\boldsymbol{\Phi_i}$ is a diagonal matrix containing non-negative values.

2. *Spatial Luminance Regularization:* First, we note that $J_1(\boldsymbol{x})$ can be rewritten as

$$J_1(\boldsymbol{x}) = \sum_{l=-P}^{P}\sum_{m=-P}^{P} \alpha^{|m|+|l|} \left\| \left(\boldsymbol{I} - \boldsymbol{S}_{\boldsymbol{x}}^{\boldsymbol{l}}\boldsymbol{S}_{\boldsymbol{y}}^{\boldsymbol{m}}\right)\boldsymbol{x_L}\right\|_1 \tag{A.21}$$

$$= \sum_{l=-P}^{P}\sum_{m=-P}^{P} \alpha^{|m|+|l|} \left\| \underbrace{\left(\boldsymbol{I} - \boldsymbol{S}_{\boldsymbol{x}}^{\boldsymbol{l}}\boldsymbol{S}_{\boldsymbol{y}}^{\boldsymbol{m}}\right)}_{\boldsymbol{A}} \underbrace{(a_R\boldsymbol{x_R} + a_G\boldsymbol{x_G} + a_B\boldsymbol{x_B})}_{\boldsymbol{x_L}} \right\|_1 \tag{A.22}$$

By noting that $d\boldsymbol{x_L} = \sum_{i=R,G,B} a_i \; d\boldsymbol{x_i}$ and then using (A.11), we have

$$\nabla_{\boldsymbol{x_i}} J_1 = a_i \sum_{l=-P}^{P}\sum_{m=-P}^{P} \alpha^{|m|+|l|} \left(\boldsymbol{I} - \boldsymbol{S}_{\boldsymbol{x}}^{-\boldsymbol{l}}\boldsymbol{S}_{\boldsymbol{y}}^{-\boldsymbol{m}}\right) \text{sign}\left(\left(\boldsymbol{I} - \boldsymbol{S}_{\boldsymbol{x}}^{\boldsymbol{l}}\boldsymbol{S}_{\boldsymbol{y}}^{\boldsymbol{m}}\right)\boldsymbol{x_L}\right) \tag{A.23}$$

where $a_R = 0.299$, $a_G = 0.587$, $a_B = 0.114$.

3. *Spatial Chromaticity Regularization:*

$$\nabla_{\boldsymbol{x_i}} J_2 = \nabla_{\boldsymbol{x_i}} \|\boldsymbol{\Lambda x_{C1}}\|_2^2 + \nabla_{\boldsymbol{x_i}} \|\boldsymbol{\Lambda x_{C2}}\|_2^2 \tag{A.24}$$

$$\boldsymbol{x_{C1}} = \sum_{i=R,G,B} c_{1,i}\boldsymbol{x_i}, \quad \boldsymbol{x_{C2}} = \sum_{i=R,G,B} c_{2,i}\boldsymbol{x_i} \tag{A.25}$$

where $c_{1,R} = 0.596$, $c_{1,G} = -0.274$, $c_{1,B} = -0.322$, $c_{2,R} = 0.211$, $c_{2,G} = -0.523$, and $c_{2,B} = 0.312$. By observing that $d\boldsymbol{x_{Cn}} = \sum_{i=R,G,B} c_{n,i} \; d\boldsymbol{x_i}$ and then using (A.11), we have

$$\nabla_{\boldsymbol{x_i}} J_2 = 2 \; c_{1,i}\boldsymbol{\Lambda}^T\boldsymbol{\Lambda x_{C1}} + 2 \; c_{2,i}\boldsymbol{\Lambda}^T\boldsymbol{\Lambda x_{C2}} \tag{A.26}$$

$$= 2 \; \boldsymbol{\Lambda}^T\boldsymbol{\Lambda} \left(c_{1,i} \; \boldsymbol{x_{C1}} + c_{2,i} \; \boldsymbol{x_{C2}}\right) \tag{A.27}$$

4. *Intercolor Dependencies:* Here we outline the derivation of the gradient with respect to the green channel, $\nabla_{\boldsymbol{x_G}}(\cdot)$, of the first term of J_3 as proposed in [10]. The gradients with respect to other channels for this and the remaining terms can be derived in a similar way.

The first term of J_3 is

$$J_{3,1} = \|\boldsymbol{x_G} \odot \boldsymbol{S}_x^l \boldsymbol{S}_y^m \boldsymbol{x_B} - \boldsymbol{x_B} \odot \boldsymbol{S}_x^l \boldsymbol{S}_y^m \boldsymbol{x_G}\|_2^2 \quad \text{(A.28)}$$

$$= \|\boldsymbol{S}_x^l \boldsymbol{S}_y^m \boldsymbol{x_B} \odot \boldsymbol{x_G} - \boldsymbol{x_B} \odot \boldsymbol{S}_x^l \boldsymbol{S}_y^m \boldsymbol{x_G}\|_2^2 \quad \text{(A.29)}$$

Now, note that the element-by-element multiplication of two column vectors can be rewritten as

$$\boldsymbol{x} \odot \boldsymbol{y} = \boldsymbol{X}\boldsymbol{y} \quad \text{(A.30)}$$

where $\boldsymbol{X}$ is a diagonal matrix containing the elements of $\boldsymbol{x}$ given by

$$\boldsymbol{X} = \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_{4r^2M^2} \end{pmatrix} \quad \text{(A.31)}$$

Hence, $J_{3,1}$ can be expressed as

$$J_{3,1} = \|\boldsymbol{X}_B^{l,m} \boldsymbol{x_G} - \boldsymbol{X_B} \boldsymbol{S}_x^l \boldsymbol{S}_y^m \boldsymbol{x_G}\|_2^2 \quad \text{(A.32)}$$

$$= \| \underbrace{\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{X_B} \boldsymbol{S}_x^l \boldsymbol{S}_y^m\right)}_{\boldsymbol{A}} \boldsymbol{x_G}\|_2^2 \quad \text{(A.33)}$$

where $\boldsymbol{X}_B^{l,m}$ is the diagonal matrix corresponding to vector $\boldsymbol{S}_x^l \boldsymbol{S}_y^m \boldsymbol{x_B}$ and $\boldsymbol{X_B}$ is the diagonal matrix corresponding to vector $\boldsymbol{x_B}$. Again, we turn our attention to (A.11) to find that

$$\nabla_{\boldsymbol{x_G}} J_{3,1} = 2\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{X_B} \boldsymbol{S}_x^l \boldsymbol{S}_y^m\right)^T \left(\boldsymbol{X}_B^{l,m} - \boldsymbol{X_B} \boldsymbol{S}_x^l \boldsymbol{S}_y^m\right) \boldsymbol{x_G} \quad \text{(A.34)}$$

$$= 2\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{S}_x^{-l} \boldsymbol{S}_y^{-m} \boldsymbol{X_B}\right) \left(\boldsymbol{X}_B^{l,m} - \boldsymbol{X_B} \boldsymbol{S}_x^l \boldsymbol{S}_y^m\right) \boldsymbol{x_G} \quad \text{(A.35)}$$

By considering the linearity property of the gradient and following a reasoning similar to the described above, the elements of the gradient of J_3 are obtained

as

$$\begin{aligned}
\nabla_{\boldsymbol{x}_R} J_3 &= 2\sum_{l=-1}^{1}\sum_{m=-1}^{1}\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{S}^{-l}\boldsymbol{S}^{-m}\boldsymbol{X}_B\right)\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{X}_B\boldsymbol{S}^{l}\boldsymbol{S}^{m}\right)\boldsymbol{x}_R \\
&\quad + \left(\boldsymbol{X}_G^{l,m} - \boldsymbol{S}^{-l}\boldsymbol{S}^{-m}\boldsymbol{X}_G\right)\left(\boldsymbol{X}_G^{l,m} - \boldsymbol{X}_G\boldsymbol{S}^{l}\boldsymbol{S}^{m}\right)\boldsymbol{x}_R \qquad \text{(A.36)} \\
\nabla_{\boldsymbol{x}_G} J_3 &= 2\sum_{l=-1}^{1}\sum_{m=-1}^{1}\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{S}^{-l}\boldsymbol{S}^{-m}\boldsymbol{X}_B\right)\left(\boldsymbol{X}_B^{l,m} - \boldsymbol{X}_B\boldsymbol{S}^{l}\boldsymbol{S}^{m}\right)\boldsymbol{x}_G \\
&\quad + \left(\boldsymbol{X}_R^{l,m} - \boldsymbol{S}^{-l}\boldsymbol{S}^{-m}\boldsymbol{X}_R\right)\left(\boldsymbol{X}_R^{l,m} - \boldsymbol{X}_R\boldsymbol{S}^{l}\boldsymbol{S}^{m}\right)\boldsymbol{x}_G \qquad \text{(A.37)} \\
\nabla_{\boldsymbol{x}_B} J_3 &= 2\sum_{l=-1}^{1}\sum_{m=-1}^{1}\left(\boldsymbol{X}_G^{l,m} - \boldsymbol{S}^{-l}\boldsymbol{S}^{-m}\boldsymbol{X}_G\right)\left(\boldsymbol{X}_G^{l,m} - \boldsymbol{X}_G\boldsymbol{S}^{l}\boldsymbol{S}^{m}\right)\boldsymbol{x}_B \\
&\quad + \left(\boldsymbol{X}_R^{l,m} - \boldsymbol{S}^{-l}\boldsymbol{S}^{-m}\boldsymbol{X}_R\right)\left(\boldsymbol{X}_R^{l,m} - \boldsymbol{X}_R\boldsymbol{S}^{l}\boldsymbol{S}^{m}\right)\boldsymbol{x}_B \qquad \text{(A.38)}
\end{aligned}$$