UNIVERSITY OF CALIFORNIA
SANTA CRUZ

A FAST AND ROBUST FRAMEWORK FOR IMAGE FUSION AND ENHANCEMENT

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Sina Farsiu

December 2005

The Dissertation of Sina Farsiu
is approved:

Professor Peyman Milanfar, Chair

Professor Ali Shakouri

Professor Michael Elad

Doctor Julian Christou

Lisa C. Sloan
Vice Provost and Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

A Fast and Robust Framework for Image Fusion and Enhancement

by

Sina Farsiu

Theoretical and practical limitations usually constrain the achievable resolution of any imaging device. The limited resolution of many commercial digital cameras resulting in aliased images are due to the limited number of sensors. In such systems, the CCD readout noise, the blur resulting from the aperture and the optical lens, and the color artifacts due to the use of color filtering arrays further degrade the quality of captured images.

Super-Resolution methods are developed to go beyond camera's resolution limit by acquiring and fusing several non-redundant low-resolution images of the same scene, producing a high-resolution image. The early works on super-resolution (often designed for grayscale images), although occasionally mathematically optimal for particular models of data and noise, produced poor results when applied to real images. On another front, single frame demosaicing methods developed to reduce color artifacts, often fail to completely remove such errors.

In this thesis, we use the statistical signal processing approach to propose an effective framework for fusing low-quality images and producing higher quality ones. Our framework addresses the main issues related to designing a practical image fusion system, namely reconstruction accuracy and computational efficiency. Reconstruction accuracy refers to the problem of designing a robust image fusion method applicable to images from different imaging systems. Advocating the use of robust L_1 norm, our general framework is applicable for optimal reconstruction of images from grayscale, color, or color filtered (CFA) cameras. The performance of our proposed method is boosted by using powerful priors and is robust to both measurement (e.g. CCD read out noise) and system noise (e.g. motion estimation error). Noting that motion estimation is often considered a bottleneck in terms of super-resolution performance, we utilize the concept of "constrained motions" for enhancing the quality of super-resolved images. We

show that using such constraints will enhance the quality of the motion estimation and therefore results in more accurate reconstruction of the HR images. We also justify some practical assumptions that greatly reduce the computational complexity and memory requirements of the proposed methods. We use efficient approximation of the Kalman Filter and adopt a dynamic point of view to the super-resolution problem. Novel methods for addressing these issues are accompanied by experimental results on simulated and real data.

Acknowledgements

This work is the result of four and a half years of close collaboration with a unique team of scientists and friends. It was their sincere assistance and support that helped me reach this milestone.

First and foremost, I would like to thank my advisor Professor Peyman Milanfar, my role model of an exceptional scientist and teacher. It was a great privilege and honor to work and study under his guidance. I would also like to thank him for his friendship, empathy, and great sense of humor (and for introducing me to French press coffee). I am grateful to my mentor Professor Michael Elad, for generously sharing his intellect, and ideas. His smiling face, patience, and careful comments were constant sources of encouragement.

I would like to thank Dr. Julian Christou, for invaluable comments and feedback throughout these years and Professor Ali Shakouri for serving on my committee and reviewing this thesis. I am grateful to all UCSC professors specially Professors Benjamin Friedlander, Roberto Manduchi, Claire Max, Hai Tao, Donald Wiberg, and Michael Issacson. As the electrical engineering GSA representative for the last few years, I had the opportunity of interacting with Dean Steve Kang; I am thankful to him and Vice Provost Lisa Sloan for going the extra mile, making the graduate studies a more pleasant experience for the UCSC graduate students. I am thankful to *mes professeurs préférés* Hervé Le Mansec, Greta Hutchison, Miriam Ellis, and Angela Elsey in the French department. I owe a lot to my former M.Sc. advisors Professors Caro Lucas and Fariba Bahrami. I would like to say a big thank-you to the Oracle of the School of Engineering Carol Mullane and the wonderful UCSC staff, specially Jodi Rieger, Carolyn Stevens, Ma Xiong, Meredith Dyer, Andrea Legg, Heidi McGough, Lynne Sheehan, Derek Pearson, David Cosby, and Marcus Thayer. Also, I would like to thank the Center for Adaptive Optics (CFAO) for supporting and funding this research.

Thanks to my friends and colleagues of many years, Ali and “the-other” Sina, for their unconditional friendship and support. Many thanks are due to Dirk and Emily for always being there for me, and to Saar for his friendship and unreserved honesty.

I thank all the handsome gentlemen of “Da Lab”, XiaoGuang, Amyn, Lior, Hiro, Mike, and Davy. Thanks to the friends and family, Maryam and the Pirnazar family, Saeedeh, Reza and Shapour, and the one and only Mehrdad for their support.

My special thanks goes to a good friend and my favorite submarine captain Reay, and to Lauren (I will never forget the yummy thanksgiving dinners). I thank my favorite director/producer David and Kelly the best lawyer in the west. Thanks to the “Grill Master” Nate for being a constant source of surprise and entertainment. I thank Marco and Teresa for the “decent” pasta nights, and of course I am grateful for the friendship of the “Catholic cake slicing theory” architect and my favorite comedy critic, Mariëlle.

Finally, my thanks goes to the beautiful city of Santa Cruz and its lovely people for being so hospitable to me.

To my parents Sohrab and Sheri, and my sister Sara for their love, support, and sacrifices.

It’s the *robustness*, stupid!

-Anonymous

Chapter 1

Introduction

On the path to designing high resolution imaging systems, one quickly runs into the problem of diminishing returns. Specifically, the imaging chips and optical components necessary to capture very high resolution images become prohibitively expensive, costing in the millions of dollars for scientific applications [5]. Hence, there is a growing interest in multi-frame image reconstruction algorithms that compensate for the shortcomings of the imaging systems. Such methods can achieve high-quality images using less expensive imaging chips and optical components by capturing multiple images and fusing them. The application of such algorithms will certainly continue to proliferate in any situation where high quality optical imaging systems cannot be incorporated or are too expensive to utilize.

A block diagram representation of the problem in hand is illustrated in Figure 1.1, where a set of images are captured by a typical imaging system (e.g. a digital camcorder). As the relative motion between the scene and the camera, the readout noise of the electronic imaging sensor (e.g. the CCD), and possibly the optical lens characteristics change through the time, each estimated image captures some unique characteristic of the underlying original image.

In this thesis, we investigate a multi-frame image reconstruction framework for fusing the information of these low-quality images to achieve an image (or a set of images) with higher quality. We develop the theory and practical algorithms with real world applications. Our proposed methods result in sharp, less noisy images with higher *spatial resolution*.

The resolution of most imaging systems is limited by their optical components. The smallest resolvable resolution of such systems empirically follows the Rayleigh limit [6], and is related to the wavelength of light and the diameter of the pinhole. The lens in optical imaging systems truncates the image spectrum in the frequency domain and further limits the resolution. In typical digital imaging systems however, it is the density of the sensor (e.g. CCD) pixels that defines the the resolution limits [1] [7].

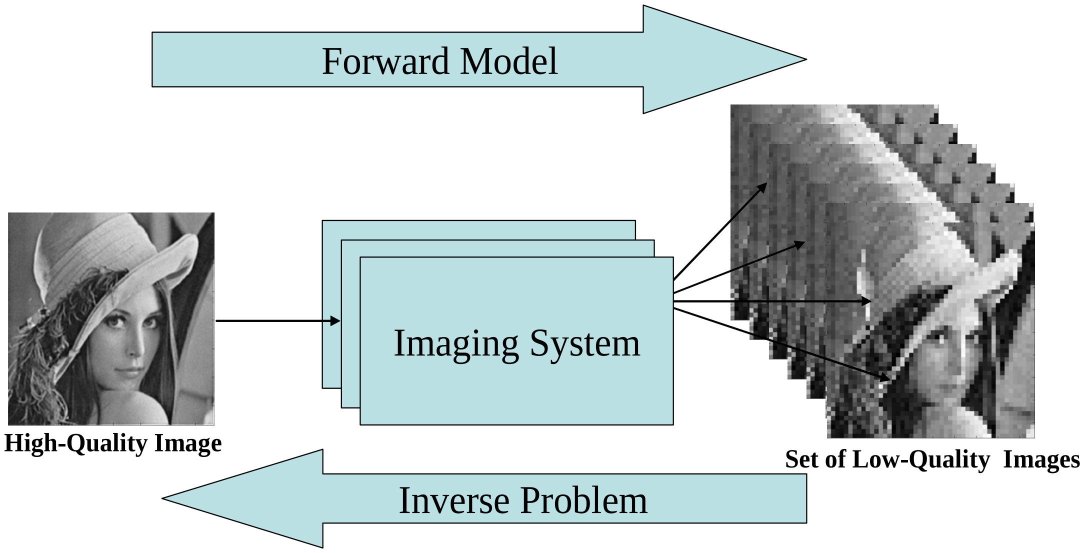


Figure 1.1: A block diagram representation of image formation and multi-frame image reconstruction in a typical digital imaging system. The forward model is a mathematical description of the image degradation process. The inverse problem addresses the issue of retrieving (or estimating) the original scene from the low-quality captured images.

An example of the multi-frame image fusion techniques is the multi-frame super-resolution, which is the main focus of this thesis. Super-resolution (SR) is the term generally applied to the problem of transcending the limitations of optical imaging systems through the

[1] Throughout this thesis, we only consider the resolution issues due to the sensor density (sampling under Nyquist limit). Although, the general framework presented here is a valuable tool for going beyond other limiting factors such as the diffraction constraints, such discussions are beyond the scope of this thesis.

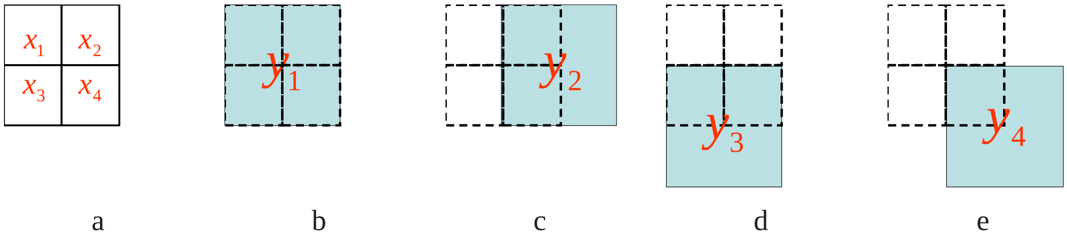


Figure 1.2: An illustrative example of the motion-based super-resolution problem. (a) A high-resolution image consisting of four pixels. (b)-(e) Low-resolution images consisting of only one pixel, each captured by subpixel motion of an imaginary camera. Assuming that the camera point spread function is known, and the graylevel of all bordering pixels is zero, the pixel values of the high-resolution image can be precisely estimated from the low-resolution images.

use of image processing algorithms, which presumably are relatively inexpensive to implement.

The basic idea behind super-resolution is the fusion of a sequence of low-resolution (LR) noisy blurred images to produce a higher resolution image. The resulting high-resolution (HR) image (or sequence) has more high-frequency content and less noise and blur effects than any of the low-resolution input images. Early works on super-resolution showed that it is the aliasing effects in the low-resolution images that enable the recovery of the high-resolution fused image, provided that a relative sub-pixel motion exists between the under-sampled input images [8].

The very simplified super-resolution experiment of Figure 1.2 illustrates the basics of the motion-based super-resolution algorithms. A scene consisting of four high-resolution pixels is shown in Figure 1.2(a). An imaginary camera with controlled subpixel motion, consisting of only one pixel captures multiple images from this scene. Figures 1.2(b)-(e) illustrate these captured images. Of course none of these low-resolution images can capture the details of the underlying image. Assuming that the point spread function (PSF) of the imaginary camera is a known linear function, and the graylevel of all bordering pixels is zero, the following equations relate the the low-resolution blurry images to the high-resolution crisper one.

That is,

$$\begin{cases} y_1 &= h_1.x_1 + h_2.x_2 + h_3.x_3 + h_4.x_4 + v_1 \\ y_2 &= 0.x_1 + h_2.x_2 + 0.x_3 + h_4.x_4 + v_2 \\ y_3 &= 0.x_1 + 0.x_2 + h_3.x_3 + h_4.x_4 + v_3 \\ y_4 &= 0.x_1 + 0.x_2 + 0.x_3 + h_4.x_4 + v_4 \end{cases},$$

where y_i 's $(i = 1, 2, 3, 4)$ are the captured low-resolution images, x_i 's are the graylevel values of the pixels in the high-resolution image, h_i's are the elements of the known PSF, and v_i's are the random additive CCD readout noise of the low-resolution frames. In cases where the additive noise is small ($v_i \simeq 0$), the above set of linear equations can be solved, obtaining the high-resolution pixel values. Unfortunately, as we shall see in the following sections the simplifying assumption made above are rarely valid in the real situations.

The experiment in Figure 1.3 shows a real example of super-resolution technology. In this experiment, a set of 26 images were captured by an OLYMPUS C-4000 camera. One of these images is shown in Figure 1.3(a). Unfortunately due to the limited number of pixels in the digital camera the details of these images are not clear, as shown in the zoomed image of Figure 1.3(b). Super-resolution helps us to reconstruct the details lost in the imaging process. The result of applying the super-resolution algorithm described in Chapter 2 is shown in Figure 1.3(c), which is a high-quality image with 16 times more pixels than any low-resolution frame (resolution enhancement factor of 4 in each direction).

Applications of the super-resolution technology include, but are not limited to:

- Industrial Applications: Designing cost-effective digital cameras, IC inspection, Designing high-quality/low-bit-rate HDTV compression algorithms.

- Scientific Imaging: Astronomy (enhancing images from telescopes), Biology (enhancing images from electronic and optical microscopes), Medical Imaging.

- Forensics and Homeland Security Applications: Enhancing images from surveillance cameras.

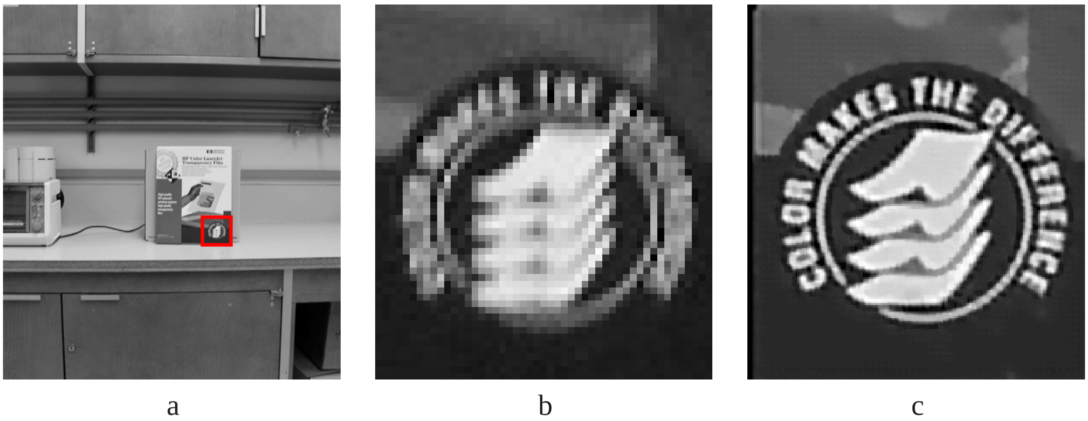


Figure 1.3: Super-resolution experiment on real world data. A set of 26 low quality images were fused resulting in a higher quality image. One captured image is shown in (a). The red square section of (a) is zoomed in (b). Super-resolved image in (c) is the high quality output image.

However, we shall see that in general, super resolution is a computationally complex and numerically ill-posed problem [2]. All this makes super-resolution one of the most appealing research areas in image processing.

1.1 Super-Resolution as an Inverse Problem

Super-resolution algorithms attempt to extract the high resolution image corrupted by the limitations of an optical imaging system. This type of problem is an example of an inverse problem, wherein the source of information (high resolution image) is estimated from the observed data (low resolution image or images). Solving an inverse problem in general requires first constructing a forward model. By far, the most common forward model for the

[2] Let $\varpi : \phi_1 \longrightarrow \phi_2$, $Y = \varpi(X)$ is said to be well-posed [9] if

1. for $Y \in \phi_2$ there exists $X \in \phi_1$, called a solution, for which $Y = \varpi(X)$ holds.
2. the solution X is unique.
3. the solution is stable with respect to perturbations in Y. This means that if $Y = \varpi(X)$and $\check{Y} = \varpi(\check{X})$ then $X \rightarrow \check{X}$ whenever $Y \rightarrow \check{Y}$.

A problem that is not well-posed is said to be ill-posed.

problem of super-resolution is linear in form

$$\underline{Y} = M\underline{X} + \underline{V} \ , \tag{1.1}$$

where $\underline{Y}$ is the measured data (single or collection of images), $\underline{X}$ is the unknown high resolution image or images, $\underline{V}$ is the random noise inherent to any imaging system.We use the underscore notation such as $\underline{X}$ to indicate a vector. In this formulation, the image is represented in vector form by scanning the 2-D image in a raster or any other scanning format[3] to 1-D.

The matrix M in the above forward model represents the imaging system, consisting of several processes that affect the quality of the estimated images. The simplest form of M is the identity matrix, which simplifies the problem at hand to a simple denoising problem. More interesting (and harder to solve) problems can be defined by considering more complex models for M. For example, to define the grey-scale super-resolution problem in Chapter 2, we consider an imaging system that consists of the blur, warp, and down-sampling processes. Moreover, addition of the color filtering process to the later model, enables us to solve for the multi-frame demosaicing problem defined in Chapter 3.

Aside from some special cases where the imaging system can be physically measured on the scene, we are bound to estimate the system matrix M from the data. In the first few chapters of this thesis (Chapters 2-4), we assume that M is given or estimated in a separate process. However, we acknowledge that such estimation is prone to errors, and design our methods considering this fact. We will discuss this in detail in the next chapter.

Armed with a forward model, a clean but practically naive solution to (1.1) can be achieved via the direct pseudo-inverse technique:

$$\underline{X} = \left(M^T M\right)^{-1} M^T \underline{Y} \ . \tag{1.2}$$

Unfortunately, the dimensions of the matrix M (as explicitly defined in the next chapters) is so large that even storing (putting aside inverting) the matrix $M^T M$ is computationally impractical.

[3] Note that this conversion is semantic and bares no loss in the description of the relation between measurements and ideal signal.

The practitioners of super-resolution usually explicitly or implicitly (e.g. the projection onto convex sets (POCS) based methods [10]) define a cost function to estimate $\underline{X}$ in an iterative fashion.This type of cost function assures a certain fidelity or closeness of the final solution to the measured data. Historically, the construction of such a cost function has been motivated from either an algebraic or a statistical perspective. Perhaps the cost function most common to both perspectives is the least-squares (LS) cost function, which minimizes the L_2 norm of the residual vector,

$$\hat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}}\, J(\underline{X}) = \underset{\underline{X}}{\text{ArgMin}}\, \|\underline{Y} - M\underline{X}\|_2^2 \ . \tag{1.3}$$

For the case where the noise $\underline{V}$ is additive white, zero mean Gaussian, this approach has the interpretation of providing the Maximum Likelihood estimate of $\underline{X}$ [11]. We shall show in this thesis that such a cost function is not necessarily adequate for super-resolution.

An inherent difficulty with inverse problems is the challenge of inverting the forward model without amplifying the effect of noise in the measured data. In the linear model, this results from the very high, possibly infinite, condition number for the model matrix M. Solving the inverse problem, as the name suggests, requires inverting the effects of the system matrix M. At best, this system matrix is ill-conditioned, presenting the challenge of inverting the matrix in a numerically stable fashion [12]. Furthermore, finding the minimizer of (1.3) would amplify the random noise $\underline{V}$ in the direction of the singular vectors (in the super-resolution case these are the high spatial frequencies), making the solution highly sensitive to measurement noise. In many real scenarios, the problem is exacerbated by the fact that the system matrix M is singular. For a singular model matrix M, there is an infinite space of solutions minimizing (1.3). Thus, for the problem of super-resolution, some form of regularization must be included in the cost function to stabilize the problem or constrain the space of solutions.

Needless to say, the choice of regularization plays a vital role in the performance of any super-resolution algorithm. Traditionally, regularization has been described from both the algebraic and statistical perspectives. In both cases, regularization takes the form of soft constraints on the space of possible solutions often independent of the measured data. This is accomplished by way of Lagrangian type penalty terms as in

$$J(\underline{X}) = \|\underline{Y} - M\underline{X}\|_2^2 + \lambda \Upsilon(\underline{X}) \ . \tag{1.4}$$

The function $\Upsilon(\underline{X})$ places a penalty on the unknown X to direct it to a better formed solution. The coefficient λ dictates the strength with which this penalty is enforced. Generally speaking, choosing λ could be either done manually, using visual inspection, or automatically using methods like Generalized Cross-Validation [13, 14], L-curve [15], and other techniques.

Tikhonov regularization[4] [11, 16, 17] is a widely employed form of regularization, which has been motivated from an analytic standpoint to justify certain mathematical properties of the estimated solution. Often, little attention, however, is given to the effects of such simple regularization on the super-resolution results. For instance, the regularization often penalizes energy in the higher frequencies of the solution, opting for a smooth and hence blurry solution. From a statistical perspective, regularization is incorporated as *a priori* knowledge about the solution. Thus, using the Maximum A-Posteriori (MAP) estimator, a much richer class of regularization functions emerges, enabling us to capture the specifics of the particular application (e.g. in [18] the piecewise-constant property of natural images are captured by modeling them as Huber-Markov random field data). Such robust methods, unlike the traditional Tikhonov penalty terms, are capable of performing adaptive smoothing based on the local structure of the image. For instance, in Chapter 2, we offer a penalty term capable of preserving the high frequency edge structures commonly found in images.

In summary, an efficient solution to the multi-frame imaging inverse problem should

1. define a forward model describing all the components of the imaging channel (such as probability density function (PDF) of additive noise, blur point spread function (PSF), relative motion vectors,...).

2. adopt proper prior information to turn the ill-posed inverse problem to a well-posed problem (regularization)

[4] Tikhonov regularization is often implemented by penalizing a high-pass filtered image by L_2 norm as formulated and explained in details in Section 2.2.3.

3. apply a method for fusing the information from multiple images which is

    (a) robust to inaccuracies in the forward model and the noise in the estimated data.

    (b) computationally efficient.

In the last two decades, many papers have been published, proposing a variety of solutions to different multi-frame image restoration related inverse problems. These methods are usually very sensitive to their assumed model of data and noise, which limits their utility. This thesis reviews some of these methods and addresses their shortcomings. We use the statistical signal processing approach to propose efficient robust image reconstruction methods to deal with different data and noise models.

1.2 Organization of this thesis

In what follows in this thesis, we study several important multi-frame image fusion/reconstruction problems under a general framework that helps us provide fast and robust solutions.

- In Chapter 2, we study the "multi-frame super-resolution" problem for grayscale images. To solve this problem, first we review the main concepts of robust estimation techniques. We justify the use of the L_1 norm to minimize the data penalty term, and propose a robust regularization technique called Bilateral Total-Variation, with many applications in diverse image processing problems. We will also justify a simple but effective image fusion technique called Shift-and-Add, which is not only very fast to implement but also gives insight to more complex image fusion problems. Finally, we propose a fast super-resolution technique for fusing grayscale images, which is robust to errors in motion and blur estimation and results in images with sharp edges.

- In Chapter 3, we focus on color images and search for an efficient method for removing color artifacts in digital images. We study the single frame "demosaicing" problem, which addresses the artifacts resulting from the color-filtering process in digital cameras. A closer look at demosaicing and super-resolution problems reveals the relation between them, and as conventional color digital cameras suffer from both low-spatial resolution and color-filtering, we optimally address them in a unified context. We propose a fast and robust hybrid method of super-resolution and demosaicing, based on a MAP estimation technique by minimizing a multi-term cost function.

- In Chapter 4, unlike previous chapters in which the final output was a single high-resolution image, we focus on producing high-resolution videos. The memory and computational requirements for practical implementation of this problem, which we call "dynamic super-resolution", are so taxing that require highly efficient algorithms. For the case of translational motion and common space-invariant blur, we propose such a method, based on a very fast and memory efficient approximation of the Kalman Filter, applicable to both grayscale and color(filtered) images.
- In Chapter 5, we address the problem of estimating the relative motion between the frames of a video sequence. In contrast to the commonly applied pairwise image registration methods, we consider global consistency conditions for the overall multi-frame motion estimation problem, which is more accurate. We review the recent work on this subject and propose an optimal framework, which can apply the consistency conditions as both hard constraints in the estimation problem, or as soft constraints in the form of stochastic (Bayesian) priors. The proposed MAP framework is applicable to virtually any motion model and enables us to develop a robust approach, which is resilient against the effects of outliers and noise.

Chapter 2

Robust Multi-Frame Super-resolution of Grayscale Images

2.1 Introduction

As we discussed in the introduction section, theoretical and practical limitations usually constrain the achievable resolution of any imaging device. In this chapter, we focus on the incoherent grayscale imaging systems and propose an effective multi-frame super-resolution method that helps improve the quality of the captured images.

A block-diagram representation of such an imaging system is illustrated in Figure 2.1, where a dynamic scene with continuous intensity distribution $X(x, y)$ is seen to be warped at the camera lens because of the relative motion between the scene and camera. The images are blurred both by atmospheric turbulence and camera lens (and CCD) by continuous point spread functions $H_{atm}(x, y)$ and $H_{cam}(x, y)$. Then they will be discretized at the CCD resulting in a digitized noisy frame Y. We represent this forward model by the following equation:

$$Y = [H_{cam}(x, y) * *F(H_{atm}(x, y) * *X(x, y))] \downarrow +V, \tag{2.1}$$

in which $**$ is the two dimensional convolution operator, F is the warping operator (projecting the scene into the camera's coordinate system), $\downarrow$ is the discretizing operator, V is the system noise and Y is the resulting discrete noisy and blurred image.

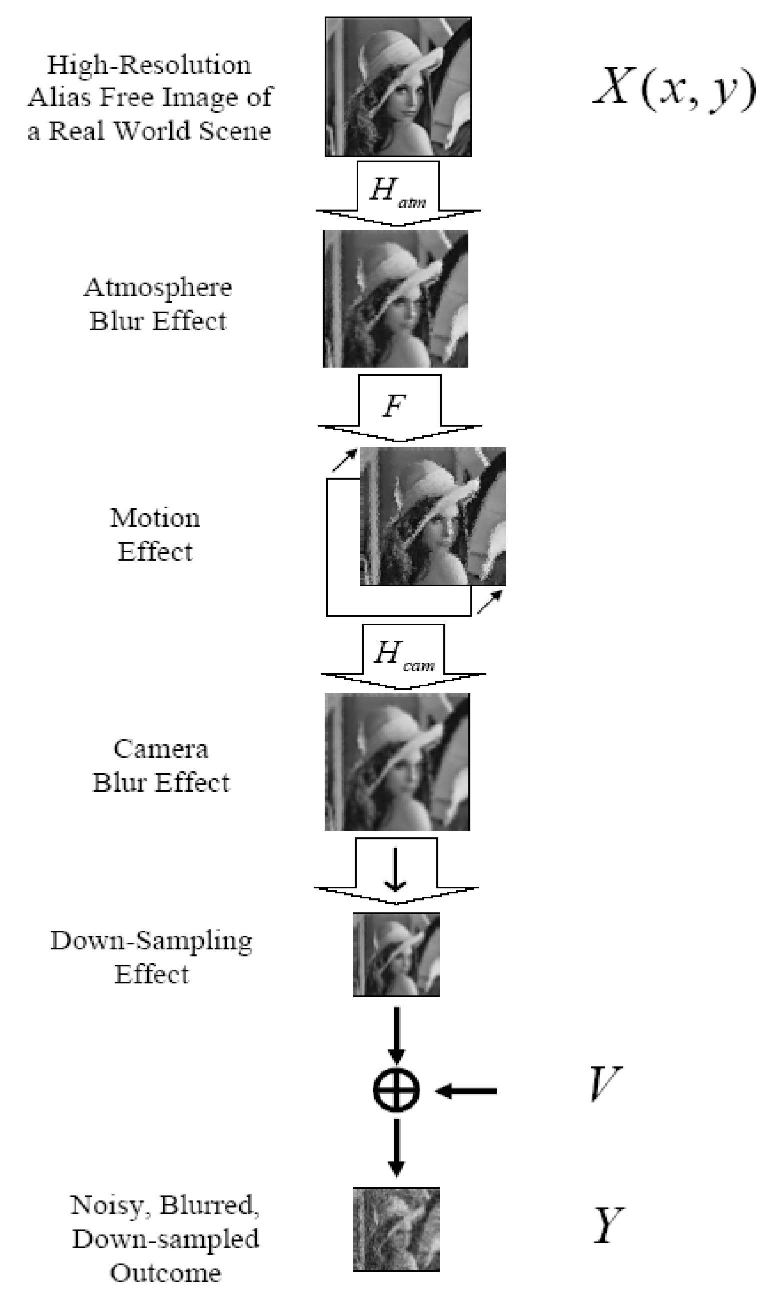


Figure 2.1: Block diagram representation of (2.1), where $X(x,y)$ is the continuous intensity distribution of the scene, V is the additive noise, and Y is the resulting discrete low-quality image.

Super-resolution is the process of combining a sequence of low-resolution noisy blurred images to produce a higher resolution image or sequence. The multi-frame super-resolution problem was first addressed in [8], where they proposed a frequency domain approach, extended by others such as [19]. Although the frequency domain methods are intuitively simple and computationally cheap, they are extremely sensitive to noise and model errors [20], limiting their usefulness. Also by design, only pure translational motion can be treated with such tools and even small deviations from translational motion significantly degrade performance.

Another popular class of methods solves the problem of resolution enhancement in the spatial domain. Non-iterative spatial domain data fusion approaches were proposed in [21], [22] and [23]. The iterative back-projection method was developed in papers such as [24] and [25]. In [26], the authors suggested a method based on the multichannel sampling theorem. In [11], a hybrid method, combining the simplicity of maximum likelihood (ML) with proper prior information was suggested.

The spatial domain methods discussed so far are generally computationally expensive. The authors in [17] introduced a block circulant preconditioner for solving the Tikhonov regularized super-resolution problem formulated in [11], and addressed the calculation of regularization factor for the under-determined case[1] by generalized cross-validation in [27]. Later, a very fast super-resolution algorithm for pure translational motion and common space invariant blur was developed in [22]. Another fast spatial domain method was recently suggested in [28], where low-resolution images are registered with respect to a reference frame defining a nonuniformly spaced high-resolution grid. Then, an interpolation method called Delaunay triangulation is used for creating a noisy and blurry high-resolution image, which is subsequently deblurred. All of the above methods assumed the additive Gaussian noise model. Furthermore, regularization was either not implemented or it was limited to Tikhonov regularization.

In recent years there has also been a growing number of *learning* based MAP meth-

[1] where the number of non-redundant low-resolution frames is smaller than the square of resolution enhancement factor. A resolution enhancement factor of r means that low-resolution images of dimension $Q_1 \times Q_2$ produce a high-resolution output of dimension $rQ_1 \times rQ_2$. Scalars Q_1 and Q_2 are the number of pixels in the vertical and horizontal axes of the low-resolution images, respectively.

ods, where the regularization-like penalty terms are derived from collections of training samples [29–32]. For example, in [31] an explicit relationship between low-resolution images of faces and their known high-resolution image is learned from a face database. This learned information is later used in reconstructing face images from low-resolution images. Due to the need for gathering a vast number of examples, often these methods are only effective when applied to very specific scenarios, such as faces or text.

Considering outliers, [1] describes a very successful robust super-resolution method, but lacks the proper mathematical justification (limitations of this robust method and its relation to our proposed method are discussed in Appendix B). Also, to achieve robustness with respect to errors in motion estimation, the very recent work of [33] has proposed an alternative solution based on modifying camera hardware. Finally, [34–36] have considered quantization noise resulting from video compression and proposed iterative methods to reduce compression noise effects in the super-resolved outcome. More comprehensive surveys of the different grayscale multi-frame super-resolution methods can be found in [7, 20, 37, 38].

Since super-resolution methods reconstruct discrete images, we use the two most common matrix notations, formulating the general continues super-resolution model of (2.1) in the pixel domain. The more popular notation used in [1, 17, 22] considers only camera lens blur and is defined as:

$$\underline{Y}(k) = D(k)H^{cam}(k)F(k)\underline{X} + \underline{V}(k) \qquad k = 1, \ldots, N \quad , \tag{2.2}$$

where the $[r^2Q_1Q_2 \times r^2Q_1Q_2]$ matrix $F(k)$ is the geometric motion operator between the discrete high-resolution frame $\underline{X}$ (of size $[r^2Q_1Q_2 \times 1]$) and the k^{th} low-resolution frame $\underline{Y}(k)$ (of size $[Q_1Q_2 \times 1]$) which are rearranged in lexicographic order and r is the resolution enhancement factor. The camera's point spread function (PSF) is modeled by the $[r^2Q_1Q_2 \times r^2Q_1Q_2]$ blur matrix $H^{cam}(k)$, and $[Q_1Q_2 \times r^2Q_1Q_2]$ matrix $D(k)$ represents the decimation operator. The $[r^2Q_1Q_2 \times 1]$ vector $\underline{V}(k)$ is the system noise and N is the number of available low-resolution frames.

Considering only atmosphere and motion blur, [28] recently presented an alternate matrix formulation of (2.1) as

$$\underline{Y}(k) = D(k)F(k)H^{atm}(k)\underline{X} + \underline{V}(k) \qquad\qquad k = 1,\ldots,N \quad . \tag{2.3}$$

In conventional imaging systems (such as video cameras), camera lens (and CCD) blur has a more important effect than the atmospheric blur (which is very important for astronomical images). In this chapter we use the model (2.2). Note that, under some assumptions which will be discussed in Section 2.2.2, blur and motion matrices commute and the general matrix super-resolution formulation from (2.1) can be rewritten as:

$$\begin{aligned} \underline{Y}(k) &= D(k)H^{cam}(k)F(k)H^{atm}(k)\underline{X} + \underline{V}(k) \\ &= D(k)H^{cam}(k)H^{atm}(k)F(k)\underline{X} + \underline{V}(k) \qquad\qquad k = 1,\ldots,N \quad . \end{aligned} \tag{2.4}$$

Defining $H(k) = H^{cam}(k)H^{atm}(k)$ merges both models into a form similar to (2.2).

In this chapter, we propose a fast and robust super-resolution algorithm using the L_1 norm, both for the regularization and the data fusion terms. Whereas the former (regularization term) is responsible for edge preservation, the latter (data fusion term) seeks robustness with respect to motion error, blur, outliers, and other kinds of errors not explicitly modeled in the fused images. We show that our method's performance is superior to what was proposed earlier in [22], [17], [1], etc. and has fast convergence. We also mathematically justify a non-iterative data fusion algorithm using a median operation and explain its superior performance.

This chapter is organized as follows: Section 2.2 explains the main concepts of robust super-resolution; subsection 2.2.2 justifies using the L_1 norm to minimize the data error term; subsection 2.2.3 justifies using our proposed regularization term; subsection 2.2.4 combines the results of the two previous sections and explains our method and subsection 2.2.5 proposes a faster implementation method. Simulations on both real and synthetic data sequences are presented in Section 2.3, and Section 2.4 concludes this chapter.

2.2 Robust Super-Resolution

2.2.1 Robust Estimation

Estimation of an unknown high-resolution image is not exclusively based on the low-resolution measurements. It is also based on many assumptions such as noise or motion models. These models are not supposed to be exactly true, as they are merely mathematically convenient formulations of some general prior information.

From many available estimators, which estimate a high-resolution image from a set of noisy low-resolution images, one may choose an estimation method which promises the optimal estimation of the high-resolution frame, based on certain assumptions on data and noise models. When the fundamental assumptions of data and noise models do not faithfully describe the measured data, the estimator performance degrades. Furthermore, existence of outliers, which are defined as data points with different distributional characteristics than the assumed model, will produce erroneous estimates. A method which promises optimality for a limited class of data and noise models may not be the most effective overall approach. Often, estimation methods which are not as sensitive to modeling and data errors may produce better and more stable robust results.

To study the effect of outliers the concept of a breakdown point has been used to measure the robustness of an algorithm. The breakdown point is the smallest percentage of outlier contamination that may force the value of the estimate outside some range [39]. For instance, the breakdown point of the simple mean estimator is zero, meaning that one single outlier is sufficient to move the estimate outside any predicted bound. A robust estimator, such as the median estimator, may achieve a breakdown equal to 0.5 (or 50 percent), which is the highest value for breakdown points. This suggests that median estimation may not be affected by data sets in which outlier contaminated measurements form less that half of all data points.

A popular family of estimators are the Maximum Likelihood type estimators (M-estimators) [40]. We rewrite the definition of these estimators in the super-resolution context as the following minimization problem:

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \rho(\underline{Y}(k), D(k)H(k)F(k)\underline{X}) \right], \tag{2.5}$$

or by an implicit equation

$$\sum_{k} \Psi(\underline{Y}(k), D(k)H(k)F(k)\underline{X}) = 0, \tag{2.6}$$

where ρ is measuring the "distance" between the model and measurements, and $\Psi(\underline{Y}(k), D(k)H(k)F(k)\underline{X}) = \frac{\partial}{\partial \underline{X}} \rho(\underline{Y}(k), D(k)H(k)F(k)\underline{X})$. The maximum likelihood estimate of $\underline{X}$ for an assumed underlying family of exponential densities $f(\underline{Y}(k), D(k)H(k)F(k)\underline{X})$ can be achieved when $\Psi(\underline{Y}(k), D(k)H(k)F(k)\underline{X}) = -\log f(\underline{Y}(k), D(k)H(k)F(k)\underline{X})$.

To find the maximum likelihood (ML) estimate of the high-resolution image, many papers such as [19], [22], [17] adopt a data model such as (2.2) and model $\underline{V}(k)$(additive noise) as white Gaussian noise. With this noise model, the least squares approach will result in the maximum likelihood estimate [41]. The least squares formulation is achieved when ρ is the L_2 norm of residual:

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \|D(k)H(k)F(k)\underline{X} - \underline{Y}(k)\|_2^2 \right]. \tag{2.7}$$

For the special case of super-resolution, based on [22], we will show in the next section, that least-squares estimation has the interpretation of being a non-robust mean estimation. As a result, least-squares based estimation of a high-resolution image, from a data set contaminated with non-Gaussian outliers, produces an image with visually apparent errors.

To appreciate this claim and study the visual effects of different sources of outliers in a video sequence, we set up the following experiments. In these experiments, four low-resolution images were used to reconstruct a higher resolution image with twice as many pixels in vertical and horizontal directions (a resolution enhancement factor of two using the least-squares approach (2.7)). Figure 2.2(a) shows the original high-resolution image and Figure 2.2(b) shows one of these low-resolution images which has been acquired by shifting Figure 2.2(a) in vertical and horizontal directions and subsampling it by factor of two (pixel replication is used to match its size with other pictures).

In the first experiment one of the four low-resolution images contained affine motion with respect to the other low-resolution images. If the model assumes translational motion, this results in a very common source of error when super-resolution is applied to real data sequences, as the respective motion of camera and the scene are seldom pure translational. Figure 2.2(c) shows this (zoomed) outlier image. Figure 2.2(d) shows the effect of this error in the motion model (shadows around Lena's hat) when the non robust least-squares approach [22] is used for reconstruction.

To study the effect of non-Gaussian noise models, in the second experiment all four low-resolution images were contaminated with salt and pepper noise. Figure 2.2(e) shows one of these low-resolution images, and Figure 2.2(f) is the outcome of the least-squares approach for reconstruction.

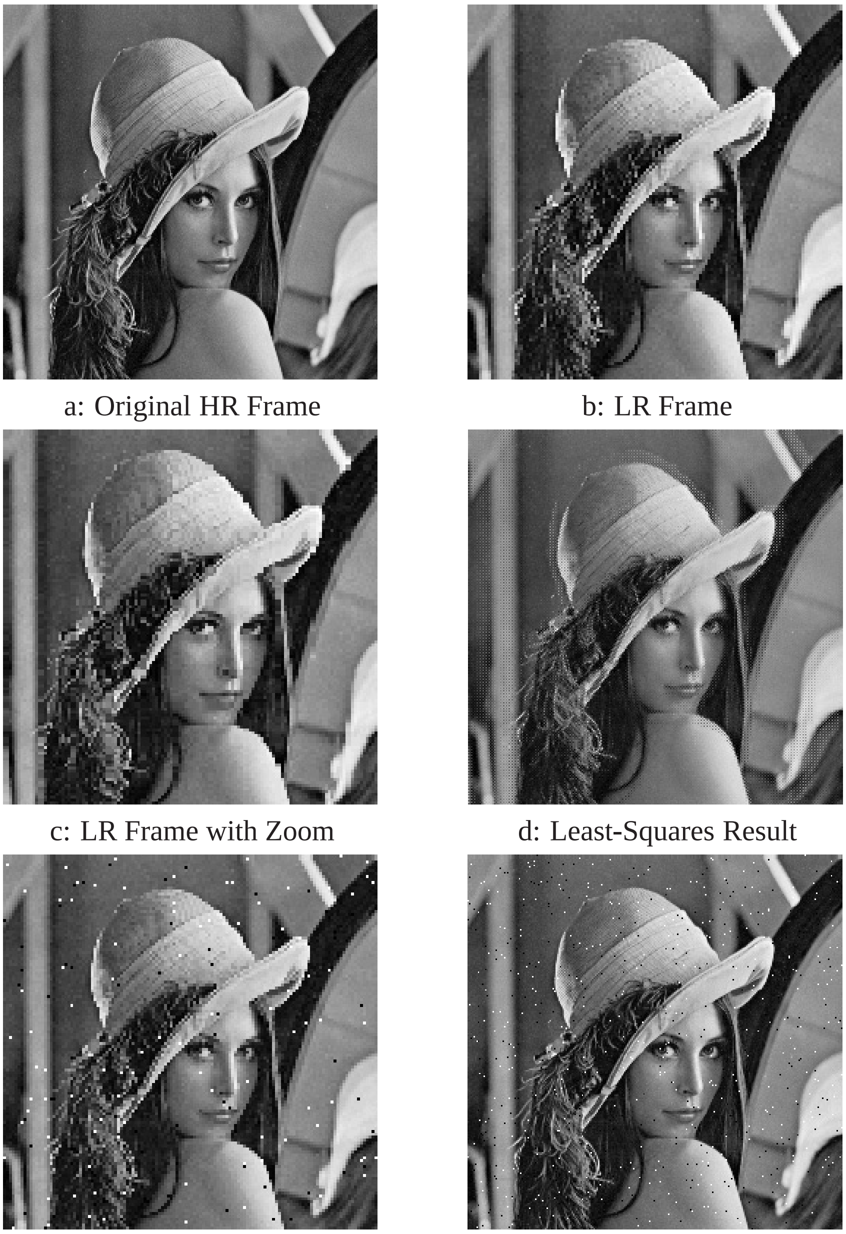
As the outlier effects are visible in the output results of least square based super-resolution methods, it seems essential to find an alternative estimator. This new estimator should have the essential properties of robustness to outliers, and fast implementation.

2.2.2 Robust Data Fusion

In subsection 2.2.1, we discussed the shortcomings of least squares based high-resolution image reconstruction. In this subsection, we study the family of $L_p, 1 \leq p \leq 2$ norm estimators. We choose the most robust estimator of this family, which results in images with the least outlier effects and show how implementation of this estimator requires minimum memory usage and is very fast.

The following expression formulates the L_p minimization criterion:

$$\widehat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \| D(k)H(k)F(k)\underline{X} - \underline{Y}(k) \|_p^p \right]. \tag{2.8}$$



a: Original HR Frame

b: LR Frame

c: LR Frame with Zoom

d: Least-Squares Result

e: LR Frame with Salt and Pepper Outlier

f: Least-Squares Result

Figure 2.2: Simulation results of outlier effects on super-resolved images. The original high-resolution image of Lena in (a) was warped with translational motion and down-sampled resulting in four images such as (b). (c) is an image acquired with downsampling and zoom (affine motion). (d) Reconstruction of these four low-resolution images with least-squares approach. (e) One of four LR images acquired by adding salt and pepper noise to set of images in (b). (f) Reconstruction of images in (e) with least-squares approach.

Note that if $p = 2$ then (2.8) will be equal to (2.7).

Considering translational motion and with reasonable assumptions such as common space-invariant PSF, and similar decimation factor for all low-resolution frames (i.e. $\forall k \quad H(k) = H \quad \& \quad D(k) = D$ which is true when all images are acquired with the same camera), we calculate the gradient of the L_p cost. We will show that L_p norm minimization is equivalent to pixelwise weighted averaging of the registered frames. We calculate these weights for the special case of L_1 norm minimization and show that L_1 norm converges to median estimation which has the highest breakpoint value.

Since H and $F(k)$ are block circulant matrices, they commute ($F(k)H = HF(k)$ and $F^T(k)H^T = H^T F^T(k)$). Therefore, (2.8) may be rewritten as:

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \| DF(k)H\underline{X} - \underline{Y}(k) \|_p^p \right]. \tag{2.9}$$

We define $\underline{Z} = H\underline{X}$. So $\underline{Z}$ is the blurred version of the ideal high-resolution image $\underline{X}$. Thus, we break our minimization problem in two separate steps:

1. Finding a blurred high-resolution image from the low-resolution measurements (we call this result $\underline{\widehat{Z}}$).

2. Estimating the deblurred image $\underline{\widehat{X}}$ from $\underline{\widehat{Z}}$

Note that anything in the null space of H will not converge by the proposed scheme. However, if we choose an initialization that has no gradient energy in the null space, this will not pose a problem (see [22] for more details). As it turns out, the null space of H corresponds to very high frequencies, which are not part of our desired solution. Note that addition of an appropriate regularization term (Section 2.2.3) will result in a well-posed problem with an empty null-space. To find $\underline{\widehat{Z}}$, we substitute $H\underline{X}$ with $\underline{Z}$:

$$\underline{\widehat{Z}} = \underset{\underline{Z}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \| DF(k)\underline{Z} - \underline{Y}(k) \|_p^p \right]. \tag{2.10}$$

The gradient of the cost in (2.10) is:

$$\begin{aligned}\underline{G}_p &= \frac{\partial}{\partial \underline{Z}}\left[\sum_{k=1}^{N} \|DF(k)\underline{Z}-\underline{Y}(k)\|_p^p\right] \\ &= \sum_{k=1}^{N} F^T(k)D^T\text{sign}(DF(k)\underline{Z}-\underline{Y}(k)) \odot |DF(k)\underline{Z}-\underline{Y}(k)|^{p-1}, \qquad (2.11)\end{aligned}$$

where operator $\odot$ is the element-by-element product of two vectors.

The vector $\widehat{\underline{Z}}$ which minimizes the criterion (2.10) will be the solution to $\underline{G}_p = \underline{0}$. There is a simple interpretation for the solution: The vector $\widehat{\underline{Z}}$ is the weighted mean of all measurements at a given pixel, after proper zero filling[2] and motion compensation.

To appreciate this fact, let us consider two extreme values of p. If $p = 2$, then

$$\underline{G}_2 = \sum_{k=1}^{N} F^T(k)D^T(DF(k)\widehat{\underline{Z}}_n - \underline{Y}(k)) = \underline{0}, \qquad (2.12)$$

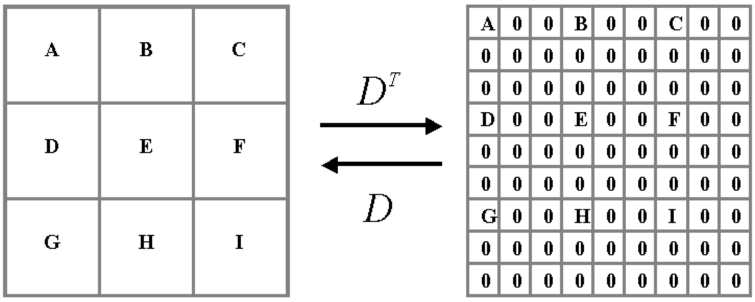
which is proved in [22] to be the pixelwise average of measurements after image registration. If $p = 1$ then the gradient term will be:

$$\underline{G}_1 = \sum_{k=1}^{N} F^T(k)D^T\text{sign}(DF(k)\widehat{\underline{Z}} - \underline{Y}(k)) = \underline{0}. \qquad (2.13)$$

We note that $F^T(k)D^T$ copies the values from the low-resolution grid to the high-resolution grid after proper shifting and zero filling, and $DF(k)$ copies a selected set of pixels in high-resolution grid back on the low-resolution grid (Figure 2.3 illustrates the effect of upsampling and downsampling matrices D^T, and D). Neither of these two operations changes the pixel values. Therefore, each element of $\underline{G}_1$, which corresponds to one element in $\widehat{\underline{Z}}$, is the aggregate of the effects of all low-resolution frames. The effect of each frame has one of the following three forms:

1. Addition of zero, which results from zero filling.
2. Addition of $+1$, which means a pixel in $\widehat{\underline{Z}}$ was larger than the corresponding contributing pixel from frame $\underline{Y}(k)$.

[2] The zero filling effect of the upsampling process is illustrated in Figure 2.3.



A	B	C
D	E	F
G	H	I

A	0	0	B	0	0	C	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
D	0	0	E	0	0	F	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
G	0	0	H	0	0	I	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

Figure 2.3: Effect of upsampling D^T matrix on a 3×3 image and downsampling matrix D on the corresponding 9×9 upsampled image (resolution enhancement factor of three). In this figure, to give a better intuition the image vectors are reshaped as matrices. In this thesis, we assume that the blurring effects of the CCD are captured by the blur matrix H, and therefore the CCD downsampling process can be modeled by a simple periodic sampling of the high-resolution image. Hence, The corresponding upsampling process is implemented as a zero filling process.

3. Addition of -1, which means a pixel in $\widehat{\underline{Z}}$ was smaller than the corresponding contributing pixel from frame $\underline{Y}(k)$.

A zero gradient state ($\underline{G}_1 = \underline{0}$) will be the result of adding an equal number of -1 and $+1$, which means each element of $\widehat{\underline{Z}}$ should be the median value of corresponding elements in the low-resolution frames. $\widehat{\underline{X}}$, the final super-resolved picture, is calculated by deblurring $\widehat{\underline{Z}}$.

So far we have shown that $p = 1$ results in pixelwise median and $p = 2$ results in pixelwise mean of all measurements after motion compensation. According to (2.11), if $1 < p < 2$ then both $\text{sign}(DF(k)\underline{Z}_n - \underline{Y}(k))$ and $|DF(k)\underline{Z}_n - \underline{Y}(k)|^{p-1}$ terms appear in $\underline{G}_p$. Therefore, when the value of p is near one, $\widehat{\underline{Z}}$ is a weighted mean of measurements, with much larger weights around the measurements near the median value, while when the value of p is near two the weights will be distributed more uniformly.

In this subsection we studied the $L_p, 1 \leq p \leq 2$ norm minimization family. As $p \longrightarrow 1$, this estimator takes the shape of median estimator, which has the highest breakpoint value, making it the most robust cost function. For the rest of this chapter, we choose L_1 to minimize the measurement error[3] (note that we left out the study of $L_p, 0 \leq p < 1$ norm

[3] L_1 norm minimization is the ML estimate of data in the presence of Laplacian noise. The statistical analysis presented in [42] and Appendices D-C justifies modeling the super-resolution noise in the presence of different

minimization family as they are not convex functions).

In the square [4] or under-determined cases, there is only one measurement available for each high-resolution pixel. As median and mean operators for one or two measurements give the same result, L_1 and L_2 norm minimizations will result in identical answers. Also in the under-determined cases certain pixel locations will have no estimate at all. For these cases, it is essential for the estimator to have an extra term, called the regularization term, to remove outliers. The next section discusses different regularization terms and introduces a robust and convenient regularization term.

2.2.3 Robust Regularization

As mentioned in Chapter 1, super-resolution is an ill-posed problem [17], [43]. For the under-determined cases (i.e. when fewer than r^2 non-redundant frames are available), there exist an infinite number of solutions which satisfy (2.2). The solution for square and over-determined [5] cases is not stable, which means small amounts of noise in the measurements will result in large perturbations in the final solution. Therefore, considering regularization in super-resolution as a means for picking a stable solution is indeed necessary. Also, regularization can help the algorithm to remove artifacts from the final answer and improve the rate of convergence. Of the many possible regularization terms, we desire one which results in high-resolution images with sharp edges and is easy to implement.

A regularization term compensates the missing measurement information with some general prior information about the desirable high-resolution solution, and is usually implemented as a penalty factor in the generalized minimization cost function (5.5):

$$\widehat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \rho(\underline{Y}(k), D(k)H(k)F(k)\underline{X}) + \lambda \Upsilon(\underline{X}) \right], \tag{2.14}$$

sources of outliers as Laplacian probability density function (PDF) rather than Gaussian PDF.

[4] where the number of non-redundant low-resolution frames is equal to the square of resolution enhancement factor.

[5] where the number of non-redundant low-resolution frames is larger than the square of resolution enhancement factor.

where λ, the regularization parameter, is a scalar for properly weighting the first term (similarity cost) against the second term (regularization cost) and Υ is the regularization cost function.

One of the most widely used regularization cost functions is the Tikhonov cost function [11], [17]:

$$\Upsilon_T(\underline{X}) = \|\Lambda \underline{X}\|_2^2, \tag{2.15}$$

where Λ is usually a high-pass operator such as derivative, Laplacian, or even identity matrix. The intuition behind this regularization method is to limit the total energy of the image (when Λ is the identity matrix) or forcing spatial smoothness (for derivative or Laplacian choices of Λ). As the noisy and edge pixels both contain high-frequency energy, they will be removed in the regularization process and the resulting reconstructed image will not contain sharp edges.

Certain types of regularization cost functions work effectively for some special types of images but are not suitable for general images. For instance Maximum Entropy regularization produces sharp reconstructions of point objects, such as star fields in astronomical images [16], however it is not applicable to natural images.

One of the most successful regularization methods for denoising and deblurring is the total variation (TV) method [44]. The total variation criterion penalizes the total amount of change in the image as measured by the L_1 norm of the magnitude of the gradient and is loosely defined as:

$$\Upsilon_{TV}(\underline{X}) = \|\nabla \underline{X}\|_1,$$

where ∇ is the gradient operator. The most useful property of total variation is that it tends to preserve edges in the reconstruction [16], [44], [45], as it does not severely penalize steep local gradients.

Based on the spirit of the total variation criterion, and a related technique called the bilateral filter (Appendix A), we introduce our robust regularizer called Bilateral Total Variation (BTV), which is computationally cheap to implement, and preserves edges. The regularizing function looks like

$$\Upsilon_{BTV}(X) = \sum_{l,m=-P}^{P} \alpha^{|m|+|l|} \|\underline{X} - S_x^l S_y^m \underline{X}\|_1, \tag{2.16}$$

where S_x^l and S_y^m are the operators corresponding to shifting the image represented by $\underline{X}$ by l pixels in the horizontal direction and m pixels in the vertical direction, respectively. This cost function in effect computes derivatives across multiple scales of resolution (as determined by the parameter P). The scalar weight $0 < \alpha < 1$ is applied to give a spatially decaying effect to the summation of the regularization term. The parameter "P" defines the size of the corresponding bilateral filter kernel. The scalar weight α, $0 < \alpha < 1$, is applied to give a spatially decaying effect to the summation of the regularization terms.

It is easy to show that this regularization method is a generalization of other popular regularization methods. If $P = \alpha = 1$, and Q_x and Q_y are the first derivative ($Q_x = I - S_x$ and $Q_y = I - S_y$) then (2.16) results in

$$\Upsilon_{BTV}(X) = \|Q_x \underline{X}\|_1 + \|Q_y \underline{X}\|_1, \tag{2.17}$$

which is suggested in [46] as a reliable and computationally efficient approximation to the Total-Variation prior [44].

To compare the performance of BTV ($P \geq 1$) to common TV prior ($P = 1$), we set up the following denoising experiment. We added Gaussian white noise of mean zero and variance 0.045 to the image in Figure 2.4(a) resulting in the noisy image of Figure 2.4(b). If $\underline{X}$ and $\underline{Y}$ represent the original and corrupted images then following (2.14), we minimized

$$\widehat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\|\underline{Y} - \underline{X}\|_2^2 + \lambda \Upsilon(\underline{X}) \right] \tag{2.18}$$

to reconstruct the original image. Tikhonov denoising resulted in Figure 2.4(c), where Λ in (2.15) was replaced by the Laplacian kernel

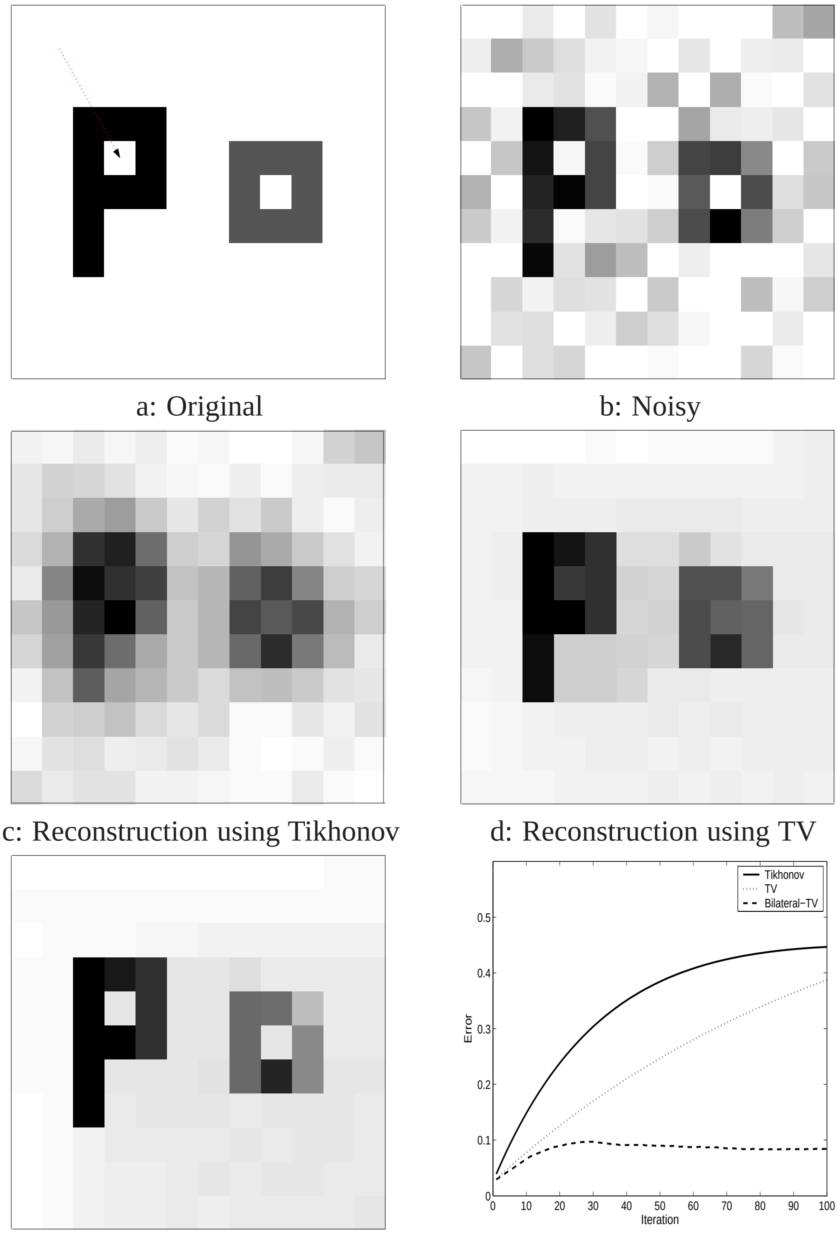
$$\Lambda = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{2.19}$$

Although a relatively large regularization factor ($\lambda = 4.5$) was chosen for this reconstruction which resulted in the loss of sharp edges, the noise has not been removed effectively. The result of using TV prior ($P = 1$, $\lambda = 0.009$) for denoising is shown in Figure 2.4(d). Figure 2.4(e) shows the result of applying BTV prior ($P = 3$, $\lambda = 0.009$)[6]. Notice the effect of each reconstruction method on the pixel indicated by an arrow in Figure 2.4(a). As this pixel is surrounded by non-similar pixels, TV prior considers it as a heavily noisy pixel, and uses the value of immediate neighboring pixels to estimate its original value. On the other hand, BTV considers a larger neighborhood. By bridging over immediate neighboring pixels, the value of similar pixels are also considered in graylevel estimation of this pixel, therefore the smoothing effect in Figure 2.4(e) is much less than Figure 2.4(d). Figure 2.4(f) compares the performance of TV and BTV denoising methods in estimating graylevel value of the arrow indicated pixel. Unlike BTV regularization, increasing the number of iterations in Tikhonov and TV regularization will result in more undesired smoothing. This example demonstrates the tendency of other regularization functionals to remove point like details from the image. The proposed regularization not only produces sharp edges but also retains point like details.

To compare the performance of our regularization method to the Tikhonov regularization method, we set up another experiment. We corrupted an image by blurring it with a Gaussian blur kernel followed by adding Gaussian additive noise. We reconstructed the image using Tikhonov and our proposed regularization terms (this scenario can be thought of as a super-resolution problem with resolution factor of one). If $\underline{X}$ and $\underline{Y}$ represent the original and corrupted images and H represents the matrix form of the blur kernel then following (2.14), we minimized

$$\widehat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\|\underline{Y} - H\underline{X}\|_2^2 + \lambda \Upsilon(X) \right] \tag{2.20}$$

[6] The criteria for parameter selection in this example (and other examples discussed in this thesis) was to choose parameters which produce visually most appealing results. Therefore to ensure fairness, each experiment was repeated several times with different parameters and the best result of each experiment was chosen as the outcome of each method. Figure 2.4(c) is an exception where we show that Tikhonov regularization fails to effectively remove noise even with a very large regularization factor.



a: Original

b: Noisy

c: Reconstruction using Tikhonov

d: Reconstruction using TV

e: Reconstruction using BTV

f: Error in gray-level value estimation

Figure 2.4: a-e: Simulation results of denoising using different regularization methods. f: Error in gray-level value estimation of the pixel indicated by arrow in (a) versus the iteration number in Tikhonov (solid line), TV (dotted line), and Bilateral TV (broken line) denoising.

to reconstruct the blurred noisy image.

Figure 2.5 shows the results of our experiment. Figure 2.5(a) shows the original image($\underline{X}$). Figure 2.5(b) is the corrupted $\underline{Y} = H\underline{X} + \underline{V}$, where $\underline{V}$ is the additive noise. Figure 2.5(c) is the result of reconstruction with Tikhonov regularization ($\Upsilon(\underline{X}) = \|\Lambda\underline{X}\|_2^2$), where Λ in (2.15) was replaced by the Laplacian kernel (2.19) and $\lambda = 0.03$. Figure 2.5(d) shows the result of applying our regularization criterion ($\Upsilon(X) = \sum_{l,m=-P}^{P} \alpha^{|m|+|l|}\|\underline{X} - S_x^l S_y^m \underline{X}\|_1$) with the following parameters: $\alpha = 0.7$, $\lambda = 0.17$ and $P = 2$. The best Mean Square Error[7] (MSE) achieved by Tikhonov regularization was 313 versus 215 for the proposed regularization. The superior edge preserving property of the bilateral prior is apparent in this example.

2.2.4 Robust Super-Resolution Implementation

In this subsection, based on the material developed in sections 2.2.2 and 2.2.3, a solution for the robust super-resolution problem will be proposed. Combining the ideas presented thus far, we propose a robust solution of the super-resolution problem as follows

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \|D(k)H(k)F(k)\underline{X} - \underline{Y}(k)\|_1 + \lambda \sum_{l,m=-P}^{P} \alpha^{|m|+|l|}\|\underline{X} - S_x^l S_y^m \underline{X}\|_1 \right]. \tag{2.21}$$

We use steepest descent to find the solution to this minimization problem:

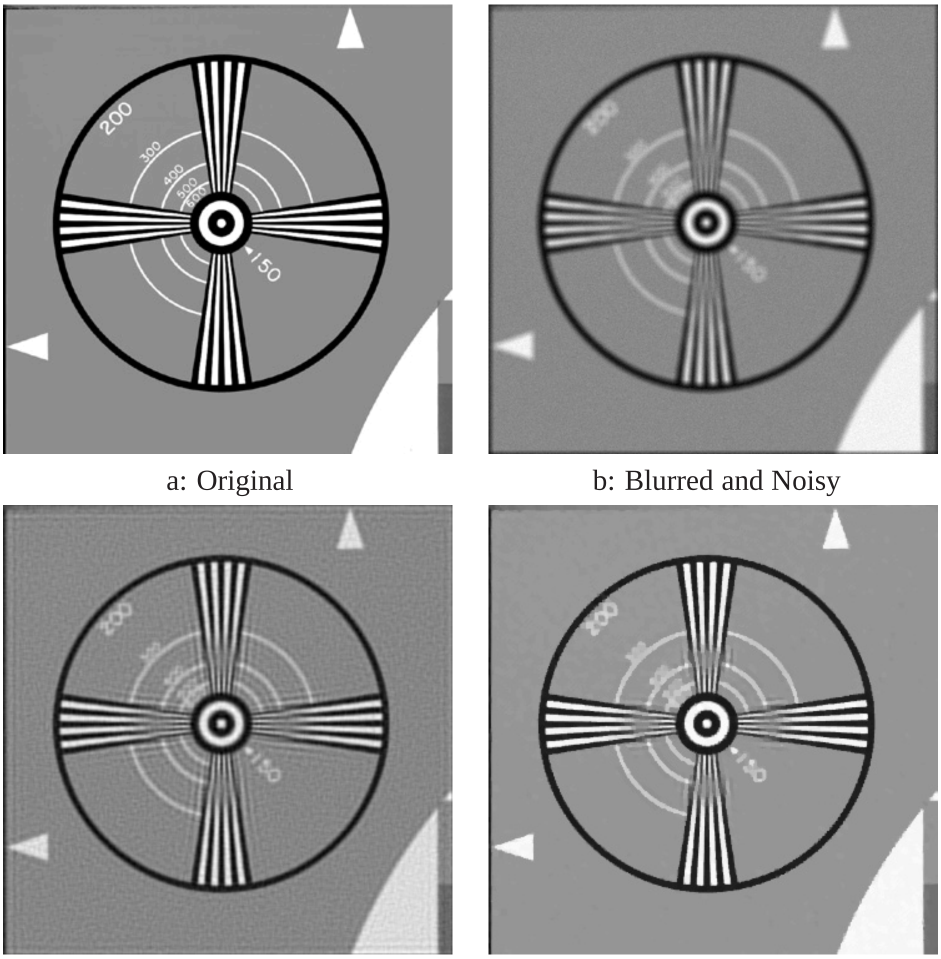
$$\underline{\widehat{X}}_{n+1} = \underline{\widehat{X}}_n - \beta \left\{ \sum_{k=1}^{N} F^T(k)H^T(k)D^T(k)\text{sign}(D(k)H(k)F(k)\underline{\widehat{X}}_n - \underline{Y}(k)) \right.$$

$$\left. + \lambda \sum_{l,m=-P}^{P} \alpha^{|m|+|l|}[I - S_y^{-m}S_x^{-l}]\text{sign}(\underline{\widehat{X}}_n - S_x^l S_y^m \underline{\widehat{X}}_n) \right\}, \tag{2.22}$$

where β is a scalar defining the step size in the direction of the gradient. S_x^{-l} and S_y^{-m} define the transposes of matrices S_x^l and S_y^m, respectively and have a shifting effect in the opposite directions as S_x^l and S_y^m.

Simulation results in Section 2.3 will show the strength of the proposed algorithm.

[7] Mean square error of an estimate is defined as MSE $= \mathrm{E}\{(\underline{\widehat{X}} - \underline{X})^2\}$, where E is the *expected value* operator, $\underline{\widehat{X}}$ is the estimate, and $\underline{X}$ is the true value of the vector to be estimated.



a: Original

b: Blurred and Noisy

c: Best Tikhonov Regularization

d: Proposed Regularization

Figure 2.5: Simulation results of deblurring using different regularization methods. The Mean Square Error (MSE) of reconstructed image using Tikhonov regularization (c) was 313. The MSE of reconstructed image using BTV (d) was 215.

The matrices F, H, D, S and their transposes can be exactly interpreted as direct image operators such as shift, blur, and decimation [47] [4]. Noting and implementing the effects of these matrices as a sequence of operators spares us from explicitly constructing them as matrices. This property helps our method to be implemented in an extremely fast and memory efficient way.

Figure 2.6 is the block diagram representation of (2.22). There, each low-resolution measurement $Y(k)$ will be compared to the warped, blurred and decimated current estimate of high-resolution frame $\widehat{X}_n$. Block G_k represents the gradient back projection operator that compares the k^{th} low-resolution image to the estimate of the high-resolution image in the n^{th} steepest descent iteration. Block $R_{m,l}$ represents the gradient of the regularization term, where the high-resolution estimate in the n^{th} steepest descent iteration is compared to its shifted version (l pixel shift in horizontal and m pixel shift in vertical directions).

Details of the blocks G_k and $R_{m,l}$ are defined in Figures 2.7(a) and 2.7(b). Block $T(PSF)$ in Figure 2.7(a) replaces the matrix $H^T(k)$ with a simple convolution. Function T flips the columns of PSF kernel in the left-right direction (that is, about the vertical axis), and then flips the rows of PSF kernel in the up-down direction (that is, about the horizontal axis)[8]. The $D^T(k)$ up-sampling block in Figure 2.7(a) can be easily implemented by filling $r-1$ zeros both in vertical and horizontal directions around each pixel (Figure 2.3). And finally the $F^T(k)$ shift-back block in Figure 2.7(a), is implemented by inverting the translational motion in the reverse direction. Note that even for the more general affine motion model a similar inverting property (though more complicated) is still valid.

Parallel processing potential of this method, which significantly increases the overall speed of implementation, can be easily interpreted from Figure 2.6, where the computation of each G_k or $R_{l,m}$ blocks may be assigned to a separate processor.

Our robust super-resolution approach also has an advantage in the computational aspects over other methods including the one proposed in [1]. In our method, an inherently robust cost function has been proposed, for which a number of computationally efficient numerical minimization methods[9] are applicable. On the contrary, [1] uses steepest descent method to minimize the non-robust L_2 norm cost function, and robustness is achieved by modifying the steepest descent method, where the median operator is used in place of summation operator in computing the gradient term of (2.12). Implementing the same scheme of substituting the summation operator with the median operator in computationally more efficient methods such

[8] If the PSF kernel has even dimensions, one extra row or column of zeros will be added to it, to make it odd size (zero columns and rows have no effect in convolution process).

[9] Such as Conjugate Gradient (CG), Preconditioned Conjugate Gradient (PCG), Jacobi, and many others.

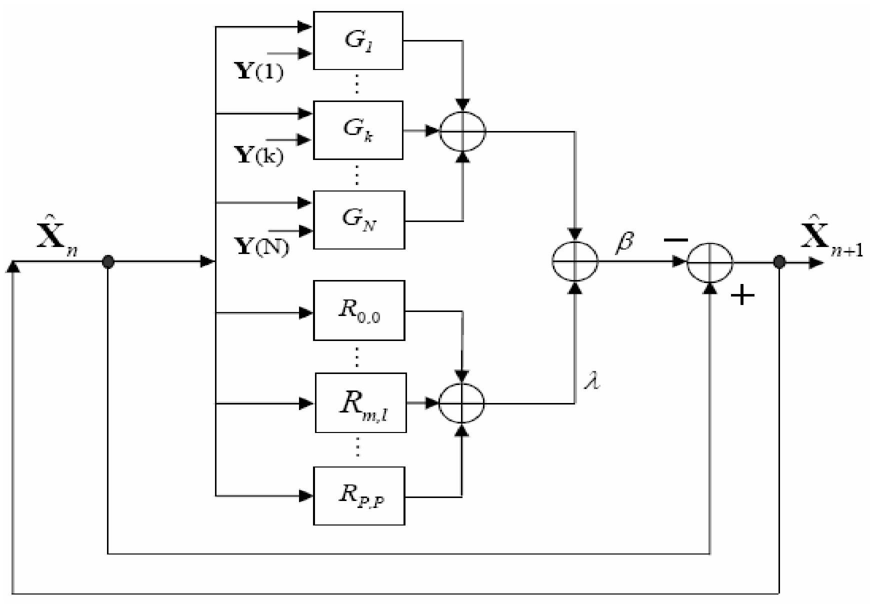
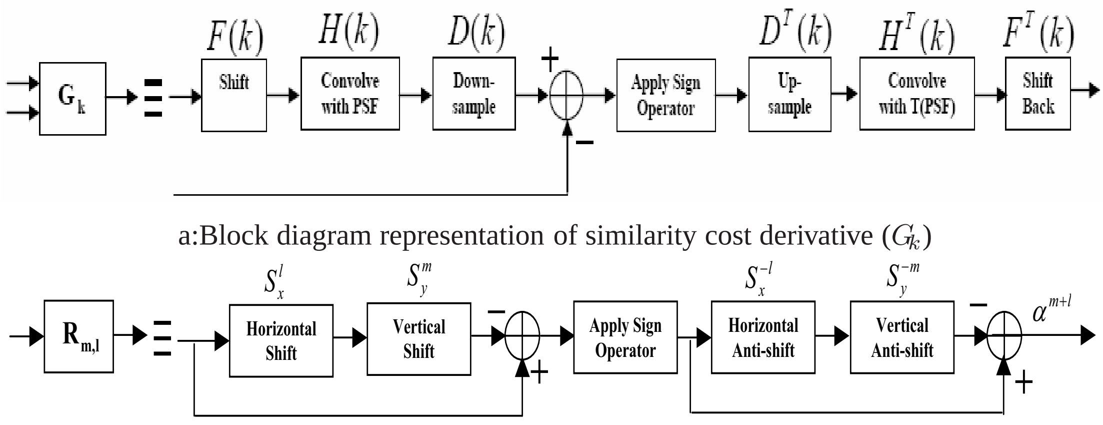


Figure 2.6: Block diagram representation of (2.22), blocks G_k and $R_{m,l}$ are defined in Figure 2.7.



a:Block diagram representation of similarity cost derivative (G_k)

b:Block diagram representation of regularization cost derivative ($R_{m,l}$)

Figure 2.7: Extended Block diagram representation of G_k and $R_{m,l}$ blocks in Figure 2.6.

as conjugate gradient is not a straightforward task and besides it is no longer guaranteed that the modified steepest descent and conjugate gradient minimization converge to the same answer.

As an example, Figure 2.8(c) and Figure 2.8(d) show the result of implementing the proposed method on the same images used to generate Figures 2.2(d), and Figure 2.2(f) (repeated in Figures 2.8(a) and 2.8(b) for the sake of comparison), respectively. The outlier effects have been reduced significantly (more detailed examples are presented in section 2.3).

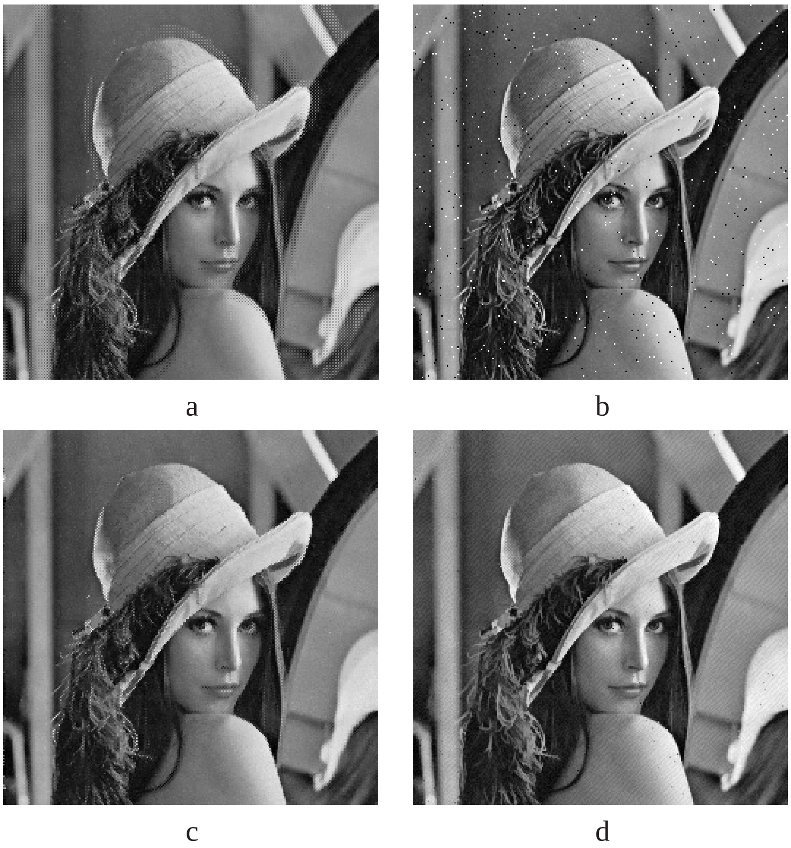


Figure 2.8: Reconstruction of the outlier contaminated images in Figure 2.2. Non-robust reconstructed images in Figures 2.2(d) and 2.2(f) are repeated in (a) and (b), respectively for the sake of comparison. The images in (c)-(d) are the robust reconstructions of the same images that was used to produce Figures (a)-(b), using equation (2.22). Note the shadow around the hat in (a) and the salt and pepper noise in (b) have been greatly reduced in (c) and (d).

In the next section we propose an alternate method to achieve further improvements in computational efficiency.

2.2.5 Fast Robust Super-Resolution Formulation

In Section 2.2.4, we proposed an iterative robust super-resolution method based on equation (2.22). Although implementation of (2.22) is very fast[10], for real-time image sequence processing, faster methods are always desirable. In this subsection, based on the interpretation of (2.13) offered in Section 2.2.2, we simplify (2.21) to achieve a faster method.

In this method, resolution enhancement is broken into two consecutive steps:

1. Non-iterative data fusion.

2. Iterative deblurring-interpolation

As we described in Section 2.2.2, registration followed by the pixelwise median operation (what we call median Shift-and-Add) results in $\underline{\widehat{Z}} = H\underline{\widehat{X}}$. Usage of the median operator for fusing low-resolution images is also suggested in [21] and [23].

The goal of the deblurring-interpolation step is finding the deblurred high-resolution frame $\underline{\widehat{X}}$. Note that for the under-determined cases not all $\underline{\widehat{Z}}$ pixel values can be defined in the data fusion step, and their values should be defined in a separate interpolation step. In this chapter unlike [21], [23] and [28], interpolation and deblurring are done simultaneously.

The following expression formulates our minimization criterion for obtaining $\underline{\widehat{X}}$ from $\underline{\widehat{Z}}$

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\|\Phi(H\underline{X} - \underline{\widehat{Z}})\|_1 + \lambda' \sum_{l,m=-P}^{P} \alpha^{|m|+|l|} \|\underline{X} - S_x^l S_y^m \underline{X}\|_1 \right], \tag{2.23}$$

where the confidence matrix Φ is a diagonal matrix with diagonal values equal to the square root of the number of measurements that contributed to make each element of $\underline{\widehat{Z}}$ (in the square case Φ is the identity matrix). So, the undefined pixels of $\underline{\widehat{Z}}$ have no effect on the high-resolution estimate $\underline{\widehat{X}}$. On the other hand, those pixels of $\underline{\widehat{Z}}$ which have been produced from numerous measurements, have a stronger effect in the estimation of the high-resolution frame $\underline{\widehat{X}}$.

As Φ is a diagonal matrix, $\Phi^T = \Phi$, and the corresponding steepest descent solution

[10] Computational complexity and memory requirement is similar to the method proposed in [25].

of minimization problem (2.23) can be expressed as

$$\begin{aligned}\underline{\widehat{X}}_{n+1} &= \underline{\widehat{X}}_n - \beta \left\{ H^T \Phi^T \text{sign}(H\underline{\widehat{X}}_n - \underline{\widehat{Z}}) \right.\\ &+ \left. \lambda' \sum_{l,m=-P}^{P} \alpha^{|m|+|l|} [I - S_y^{-m} S_x^{-l}] \text{sign}(\underline{\widehat{X}}_n - S_x^l S_y^m \underline{\widehat{X}}_n) \right\}. \qquad (2.24)\end{aligned}$$

Decimation and warping matrices (D and F) and summation of measurements are not present anymore, which makes the implementation of (2.24) much faster than (2.22). Note that physical construction of matrix Φ is not necessary as it can be implemented as a mask matrix with the size equal to that of image $\mathbf{X}$.

2.3 Experiments

In this section we compare the performance of the resolution enhancement algorithms proposed in this chapter to existing resolution enhancement methods. The first example is a controlled simulated experiment. In this experiment we create a sequence of low-resolution frames by using one high-resolution image (Figure 2.9(a)). First we shifted this high-resolution image by a pixel in the vertical direction. Then to simulate the effect of camera PSF, this shifted image was convolved with a symmetric Gaussian low-pass filter of size 4×4 with standard deviation equal to one. The resulting image was subsampled by the factor of 4 in each direction. The same approach with different motion vectors (shifts) in vertical and horizontal directions was used to produce 16 low-resolution images from the original scene. We added Gaussian noise to the resulting low-resolution frames to achieve SNR equal[11] to 18dB. One of these low-resolution frames is presented in Figure 2.9(b). To simulate the errors in motion estimation, a bias equal to one pixel shift in the low-resolution grid (or 4 pixel shift in the high-resolution grid) was intentionally added to the known motion vectors of the last three low-

[11] Signal to noise ratio (SNR) is defined as $10\log_{10}\frac{\sigma^2}{\sigma_n^2}$, where σ^2, σ_n^2 are variance of a clean frame and noise, respectively

Frame Number	1	2	3	4	5	6	7	8
Motion in X-Direction	0	0.25	0.5	0.75	0	0.25	0.5	0.75
Motion in Y-Direction	0	0	0	0	0.25	0.25	0.25	0.25
Frame Number	9	10	11	12	13	14	15	16
Motion in X-Direction	0	0.25	0.5	0.75	0	0.25	0.5	0.75
Motion in Y-Direction	0.5	0.5	0.5	0.5	0.75	0.75	0.75	0.75

Table 2.1: The true motion vectors (in the low-resolution grid) used for creating the low-resolution frames in the experiment presented in Figure 2.9.

Frame Number	1	2	3	4	5	6	7	8
Motion in X-Direction	0	0.25	0.5	0.75	0	0.25	0.5	0.75
Motion in Y-Direction	0	0	0	0	0.25	0.25	0.25	0.25
Frame Number	9	10	11	12	13	14	15	16
Motion in X-Direction	0	0.25	0.5	0.75	0	-0.75	-0.5	-0.25
Motion in Y-Direction	0.5	0.5	0.5	0.5	0.75	-0.25	-0.25	-0.25

Table 2.2: The erroneous motion vectors (in the low-resolution grid) used for reconstructing the high-resolution frames of the experiments presented in Figure 2.9.

resolution frames. The correct and erroneous motion vectors are shown in Table 2.1 and Table 2.2, respectively.

The result of implementing the non-iterative resolution enhancement method described in [22] is shown in Figure 2.9(c). It is not surprising to see the motion error artifacts in the high-resolution frame as the high-resolution image is the result of zero-filling, shifting and adding the low-resolution measurements. Deblurring this result with the Wiener method[12] (Figure 2.9(d)) does not remove these artifacts, of course. For reference, Figure 2.9(e) shows the result of applying an iterative method based on minimizing the L_2 norm, both for the residual and the regularization terms. The following equation describes this minimization criterion

$$\underline{\widehat{X}}=\text{ArgMin}\left[\sum_{k=1}^{N}\|D(k)H(k)F(k)\underline{X}-\underline{Y}(k)\|_2^2+\lambda\|\Lambda\underline{X}\|_2^2\right], \tag{2.25}$$

in which Λ is defined in (2.19) and regularization factor λ was chosen to be 0.4. As the L_2

[12] The Wiener deblurring is implemented by convolving a linear spatially-invariant kernel, known as Wiener Filter, with the blurred image. Winer Filter, often estimated in the frequency domain, is the linear kernel that minimizes the MSE of the difference between the ideal image and the restored one [16].

norm is not robust to motion error, motion artifacts are still visible in the result. Note that the relatively high regularization factor , chosen to reduce the motion artifact, has resulted in a blurry image.

The robust super-resolution method which was proposed in [1] resulted in Figure 2.9(f). Figure 2.9(g) was obtained by simply adding the regularization term defined in (2.25) to the proposed method of [1] which is far better than the L_2 approach, yet exhibiting some artifacts. Figure 2.9(h) shows the implementation of the proposed method described in Section 2.2.4. The selected parameters for this method were as follows: $\lambda = 0.005$, $P = 2$, $\beta = 110$, $\alpha = 0.6$. Figure 2.9(i) shows the implementation of the fast method described in Section 2.2.5. The selected parameters for this method were as follows: $\lambda' = 0.08$, $P = 2$, $\beta = 1$, $\alpha = 0.6$. Comparing Figure 2.9(h) and 2.9(i) to other methods, we notice not only our method has removed the outliers more efficiently, but also it has resulted in sharper edges without any ringing effects.

Our second example is a real infrared camera image sequences with no known outliers; courtesy of B. Yasuda and the FLIR research group in the Sensors Technology Branch, Wright Laboratory, WPAFB, OH. We used eight low-resolution frames of size $[64 \times 64]$ in our reconstruction to get resolution enhancement factor of four (Figure 2.10(a) shows one of the input low-resolution images)[13]. Figure 2.10(b) of size $[256 \times 256]$ shows the cubic spline interpolation of Figure 2.10(a) by factor of four . The (unknown) camera PSF was assumed to be a 4×4 Gaussian kernel with standard deviation equal to one. We used the method described in [48] to computed the motion vectors. L_2 norm reconstruction with Tikhonov regularization (2.25) result is shown in Figure 2.10(c) where Λ is defined in (2.19) and regularization factor λ was chosen to be 0.1. Figure 2.10(d) shows the implementation of (2.22) with the following parameters $\lambda = 0.006$, $P = 2$, $\beta = 81$, and $\alpha = 0.5$. Although modeling noise in these frames as additive Gaussian is a reasonable assumption, our method achieved a better result than the best L_2 norm minimization.

[13] Note that this is an under-determined scenario.

Our third experiment is a real compressed sequence of 20 images (containing translational motion) from a commercial surveillance video camera; courtesy of Adyoron Intelligent Systems Ltd., Tel Aviv, Israel. Figure 2.11(a) is one of these low-resolution images (of size $[76 \times 66]$) and Figure 2.11(b) is the cubic spline interpolation of this image by factor of three (of size $[228 \times 198]$). We intentionally rotated five frames of this sequence (rotation from 20° to 60°) out of position, creating a sequence of images with relative affine motion. The (unknown) camera PSF was assumed to be a 5×5 Gaussian kernel with standard deviation equal to two. We used the method described in [48] to computed the motion vectors with translational motion assumption. The error in motion modeling results in apparent shadows in L_2 norm reconstruction with Tikhonov regularization (Figure 2.11(c)) where Λ is defined in (2.19) and regularization factor λ was chosen to be 0.5. These shadows are removed in Figure 2.11(d), where the method described in Section 2.2.4 (2.22) was used for reconstruction with the following parameters $\lambda = 0.003$, $P = 2$, $\beta = 50$, and $\alpha = 0.7$.

Our final experiment is a factor of three resolution enhancement of a real compressed image sequence captured with a commercial webcam (3Com, Model No.3718). The (unknown) camera PSF was assumed to be a 3×3 Gaussian kernel with standard deviation equal to 1. In this sequence, two separate sources of motion were present. First, by shaking the camera a global motion was created for each individual frame. Second, an Alpaca statue was independently moved in to ten frames out of the total 55 input frames. One of the low-resolution input images (of size $[32 \times 65]$) is shown in Figure 2.12(a). Cubic spline interpolation of Figure 2.12(a) by factor of three is shown in Figure 2.12(b). Figure 2.12(c) and Figure 2.12(d) (of size $[96 \times 195]$) are the shift and add results using mean and median operators (minimizing$\widehat{\underline{Z}}$ in (2.10) with $p = 2$ and $p = 1$, respectively). Note that the median operator has lessened the (shadow) artifacts resulting from the Alpaca motion. L_2 norm reconstruction with Tikhonov regularization (2.25) results in Figure 2.12(e), where Λ is defined in (2.19) and regularization factor λ was chosen to be one. Figure 2.12(f) is the result of minimizing the following cost function

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\sum_{k=1}^{N} \|D(k)H(k)F(k)\underline{X} - \underline{Y}(k)\|_2^2 + \lambda \sum_{l,m=-P}^{P} \alpha^{|m|+|l|} \|\underline{X} - S_x^l S_y^m \underline{X}\|_1 \right],$$

where L_2 norm minimization of data error term is combined with Bilateral TV regularization with the following parameters $\lambda = 0.1$, $P = 2$, $\alpha = 0.7$, and $\beta = 70$ (steepest descent step size). Note that the artifacts resulting from the motion of the Alpaca statue is visible in Figures 2.12(d)-(g). The result of using the robust super-resolution method proposed in [1] is shown in Figure 2.12(h). Implementation of the method described in Section 2.2.4 equation (2.22) with the following parameters $\lambda = 0.003$, $P = 2$, $\beta = 30$, and $\alpha = 0.7$ resulted in Figure 2.12(i), with the least outlier effect. And finally implementation of the fast method described in Section 2.2.5 (2.24) with the following parameters $\lambda' = 0.04$, $P = 2$, $\beta = 1$, and $\alpha = 0.7$ resulted in Figure 2.12(j), which is very similar to the result in Figure 2.12(i).

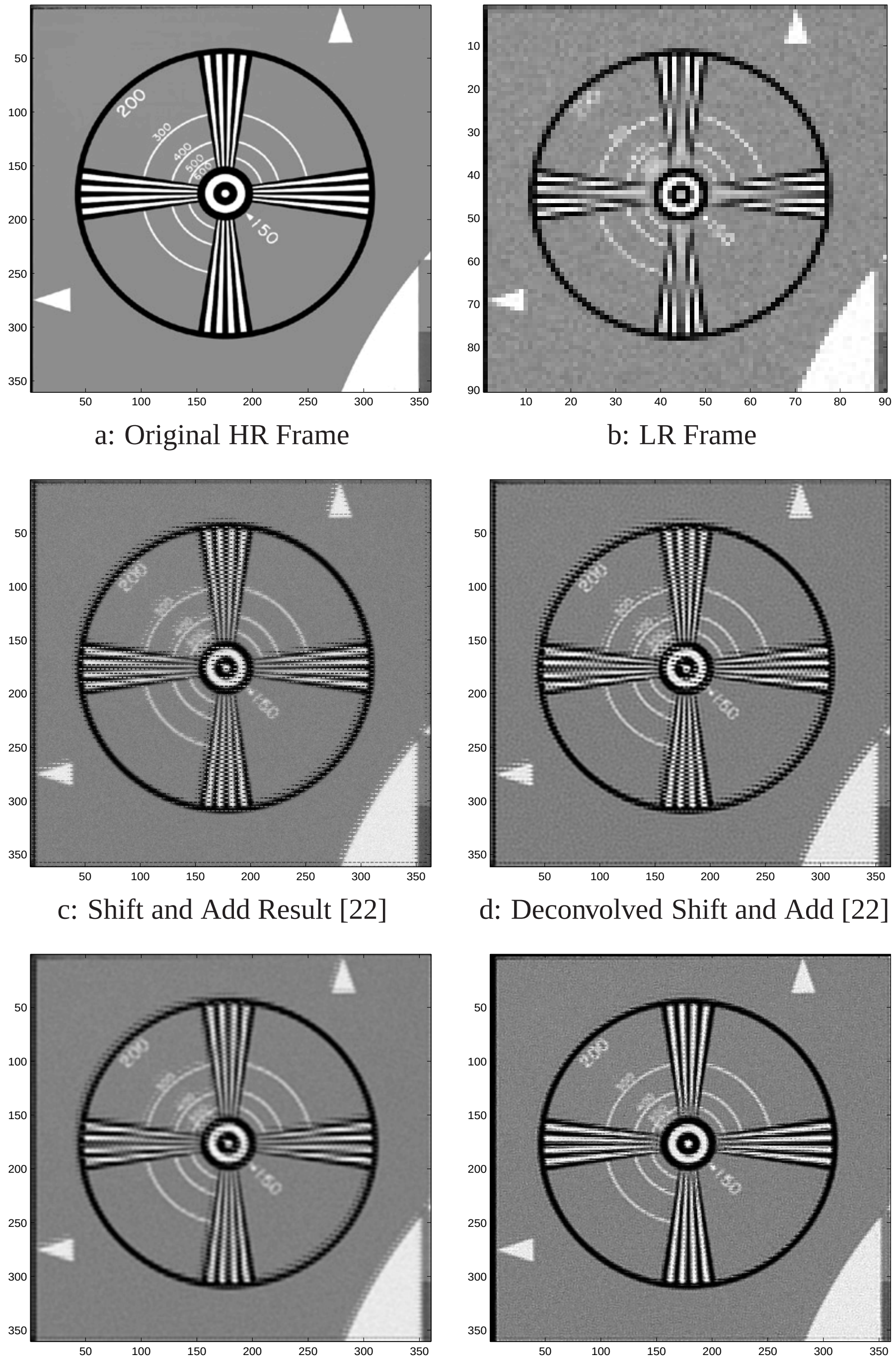
2.4 Summary and Discussion

In this chapter, we presented an algorithm to enhance the quality of a set of noisy blurred images and produce a high-resolution image with less noise and blur effects. We presented a robust super-resolution method based on the use of L_1 norm both in the regularization and the measurement terms of our penalty function. We showed that our method removes outliers efficiently, resulting in images with sharp edges. Even for images in which the noise followed the Gaussian model, L_1 norm minimization results were as good as L_2 norm minimization results, which encourages using L_1 norm minimization for any data set. The proposed method was fast and easy to implement.

We also proposed and mathematically justified a very fast method based on pixelwise "Shift-and-Add" and related it to L_1 norm minimization when relative motion is pure translational, and PSF and decimation factor is common and space-invariant in all low-resolution images. Note that the mathematical derivation of the proposed shift and add method was independent of the constraint over decimation factor, but we included it as this constraint distinguishes super-resolution from the more general problem of multi-scale image fusion. In the presented experiments, we rounded the displacements in the high-resolution grid so that $F(k)$ applies only integer translations. This will not pose a problem as the rounding is done only on the high-resolution grid [22]. However, we could as well shift the low-resolution images by subpixel motion (e.g. using spline interpolation) as the justification given in Section 2.2.2 and the formulation in (2.23) is general enough for both cases.

Analysis of the rate of convergence of the steepest descent method is only possible for simplistic cases such as minimizing a quadratic function. Considering quantized images, L_1 norm minimization, and regularization terms make such analysis much harder. We have observed that only 5-20 iterations are required for convergence to the desired solution, where the initialization and the type of involved images play a vital role in determining the required number of iterations. The outcome of the speed-up method of Section 2.2.5 is a very good initialization guess for the more general case of Section 2.2.4.

Although "cross validation" can be used to determine the parameter values [27], implementing this for the L_1 norm is rather more difficult and computationally expensive. Parameters like P can also be learned using a learning algorithm, however such an approach is outside the scope of this chapter. We have found that setting P to 2 or 3 works well; using higher values for P will be time consuming while not very useful in terms of producing higher quality results.



a: Original HR Frame

b: LR Frame

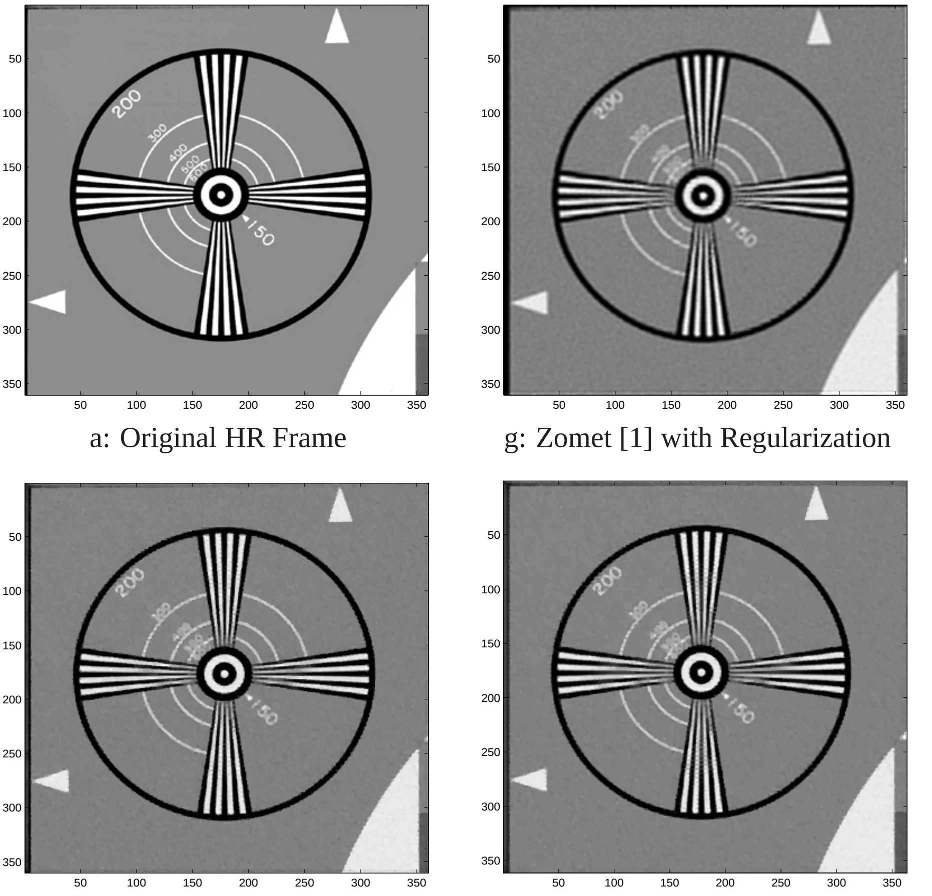
c: Shift and Add Result [22]

d: Deconvolved Shift and Add [22]

e: L_2 + Tikhonov

f: Zomet Method [1]

Figure 2.9: Controlled simulation experiment. Different resolution enhancement methods ($r = 4$) are applied to the Figure (b).



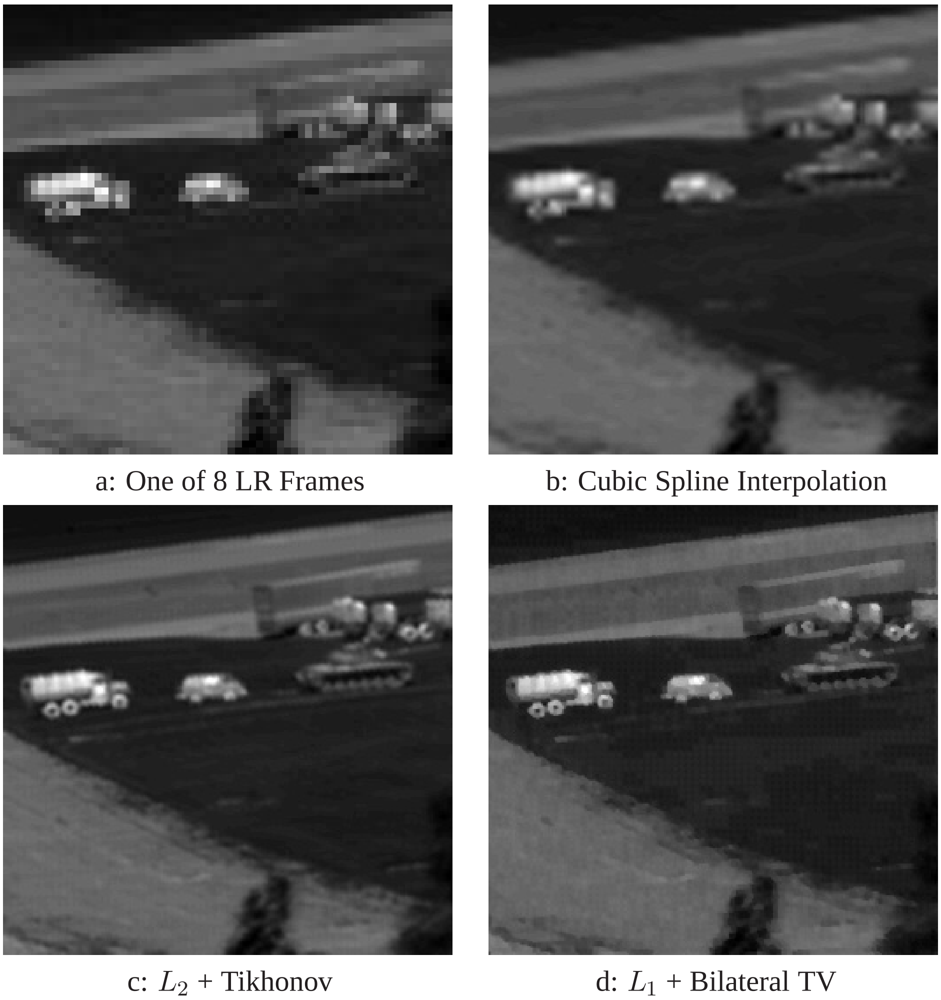
a: Original HR Frame

g: Zomet [1] with Regularization

h: L_1 + Bilateral TV

i: Median Shift and Add + Bilateral TV

Figure 2.9: Controlled simulation experiment. Different resolution enhancement methods ($r = 4$) are applied to the Figure (b).



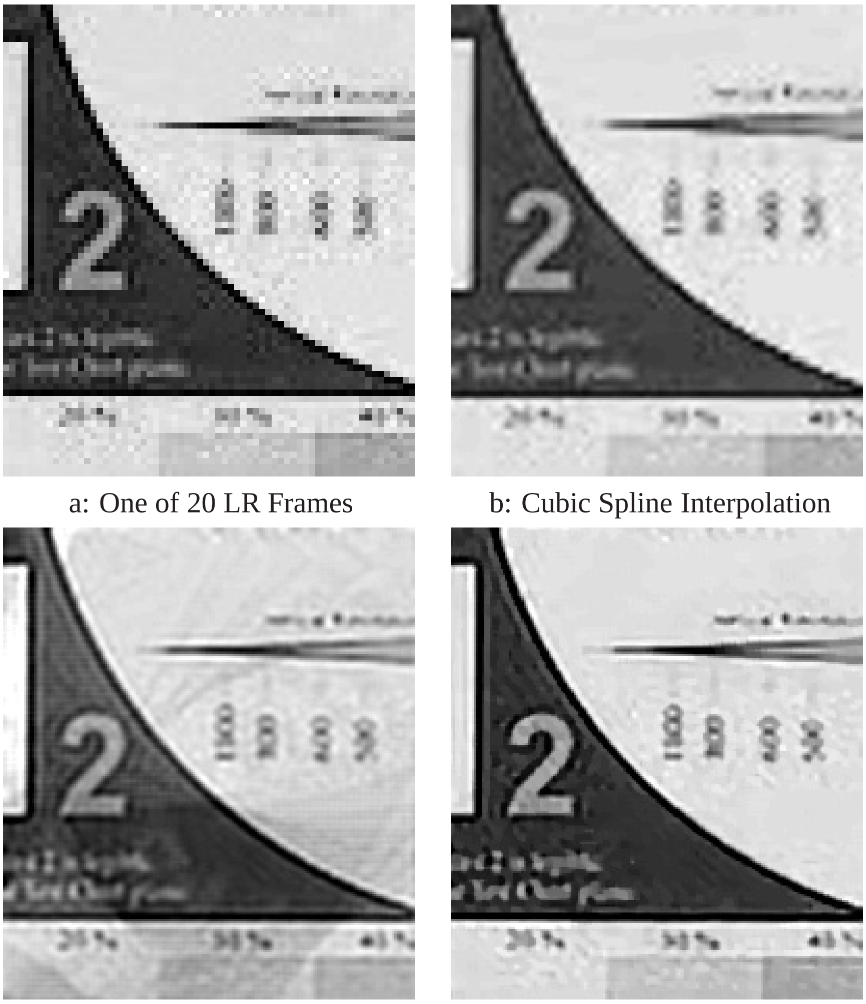
a: One of 8 LR Frames

b: Cubic Spline Interpolation

c: L_2 + Tikhonov

d: L_1 + Bilateral TV

Figure 2.10: Results of different resolution enhancement methods ($r = 4$) applied to Tank sequence.



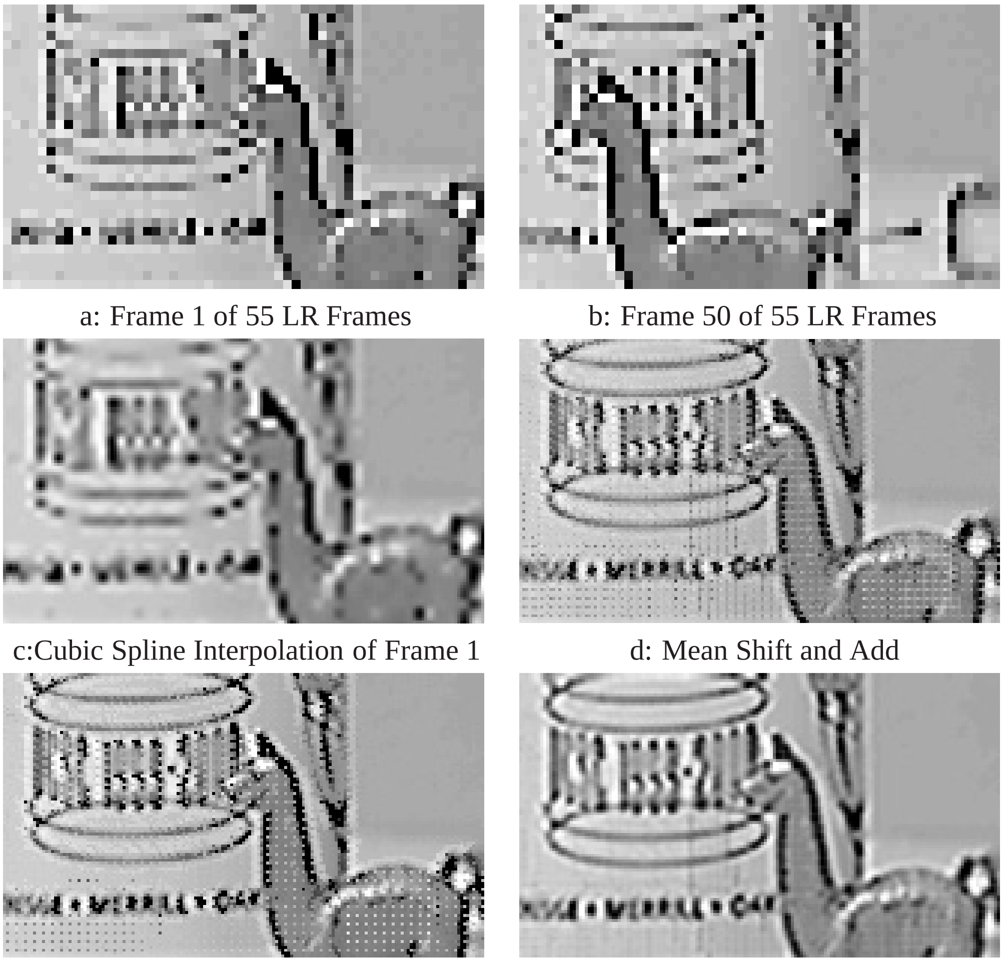
a: One of 20 LR Frames

b: Cubic Spline Interpolation

c: L_2 + Tikhonov

d: L_1 + Bilateral TV

Figure 2.11: Results of different resolution enhancement methods ($r = 4$) applied to Adyoron test sequence.



a: Frame 1 of 55 LR Frames

b: Frame 50 of 55 LR Frames

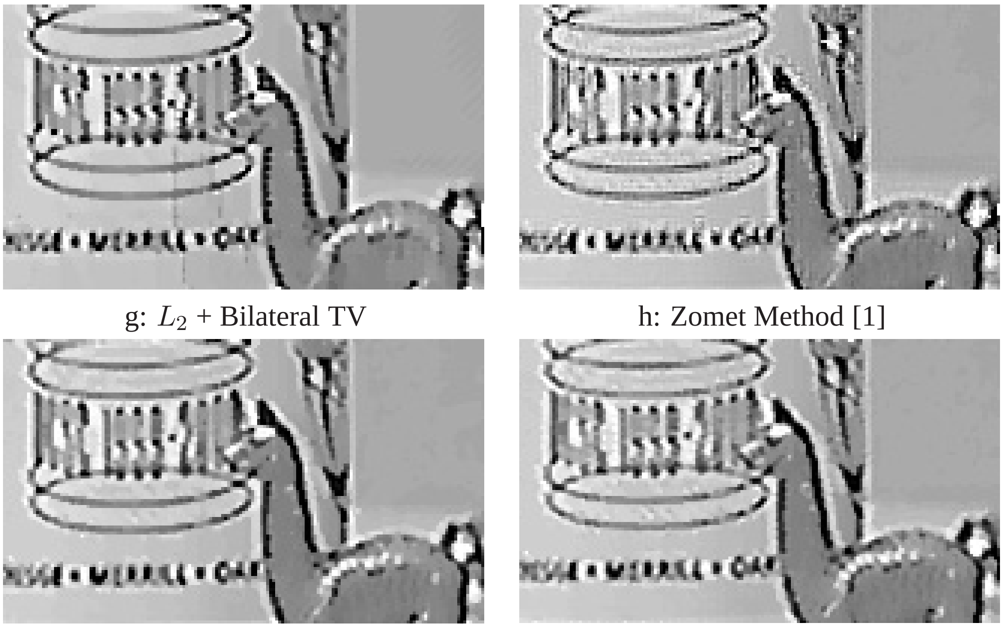
c:Cubic Spline Interpolation of Frame 1

d: Mean Shift and Add

e: Median Shift and Add

f: L_2 + Tikhonov

Figure 2.12: Results of different resolution enhancement methods ($r = 3$) applied to the Alpaca sequence. Outlier effects are apparent in the non-robust reconstruction methods.



g: L_2 + Bilateral TV

h: Zomet Method [1]

i: L_1 + Bilateral TV

j: Median Shift and Add + Bilateral TV

Figure 2.12: Results of different resolution enhancement methods applied to the Alpaca sequence. Outlier effects are apparent in the non-robust reconstruction method (g). The shadow of the Alpaca is removed in the robust reconstruction methods of (h),(i), and (j).

Chapter 3

Multi-Frame Demosaicing and Color Super-Resolution

3.1 Introduction

In digital image processing, two reconstruction problems have been studied and solved independently - super-resolution and demosaicing. The former (as studied in the previous chapter) refers to the limited number of pixels and the desire to go beyond this limit using several exposures. The latter refers to the color-filtering applied on a single CCD array of sensors on most cameras, that measures a subset of R (red), G (green), and B (blue) values, instead of a full RGB field[1]. It is natural to consider these problems in a joint setting because both refer to resolution limitations at the camera. Also, since the measured images are mosaiced, solving the super-resolution problem using pre-processed (demosaiced) images is sub-optimal and hence inferior to a single unifying solution framework. In this chapter we propose a fast and robust method for joint multi-frame demosaicing and color super-resolution.

The organization of this chapter is as follows. In Section 3.2 we review the super-resolution and demosaicing problems and the inadequacy of independent solutions for them.

[1] Three CCD cameras which measure each color field independently tend to be relatively more expensive.

In Section 3.3 we formulate and analyze the general mathematical model of the imaging systems we consider here. We also formulate and review the basics of the MAP estimator, robust data fusion, and regularization methods. Armed with material developed in earlier sections, in Section 3.4 we present and formulate our joint multi-frame demosaicing and color-super-resolution method. In Section 3.5 we review two related methods of multi-frame demosaicing. Experiments on both synthetic and real data sequences are given in Section 3.6 and concluding remarks are presented in Section 3.7.

3.2 An overview of super-resolution and demosaicing problems

In this section, we study and review some of the previous work on super-resolution and demosaicing problems. We show the inefficiency of independent solutions for these problems and discuss the obstacles to designing a unified approach for addressing these two common shortcomings of digital cameras.

3.2.1 Super-Resolution

Digital cameras have a limited spatial resolution, dictated by their utilized optical lens and CCD array. Surpassing this limit can be achieved by acquiring and fusing several low-resolution images of the same scene, producing high-resolution images; this is the basic idea behind super-resolution techniques [20, 38, 49] as studied in the previous chapter.

Note that almost all super-resolution methods to date have been designed to increase the resolution of a single channel (grayscale or monochromatic) image. A related problem, color SR, addresses fusing a set of previously demosaiced color low-resolution frames to enhance their spatial resolution. To date, there is very little work addressing the problem of color SR. The typical solution involves applying monochromatic SR algorithms to each of the color channels independently [50, 51], while using the color information to improve the accuracy of motion estimation. Another approach is transforming the problem to a different color space, where chrominance layers are separated from luminance, and SR is applied only to the luminance channel [25]. Both of these methods are sub-optimal as they do not fully exploit the correlation across the color bands.

In Section 3.6 we show that ignoring the relation between different color channels will result in color artifacts in the super-resolved images. Moreover, as we will advocate later in this chapter, even a proper treatment of the relation between the color layers is not sufficient for removing color artifacts if the measured images are mosaiced. This brings us to the description of the demosaicing problem.

3.2.2 Demosaicing

A color image is typically represented by combining three separate monochromatic images. Ideally, each pixel reflects three data measurements; one for each of the color bands[2]. In practice, to reduce production cost, many digital cameras have only one color measurement (red, green, or blue) per pixel [3]. The detector array is a grid of CCDs, each made sensitive to one color by placing a color-filter array (CFA) in front of the CCD. The Bayer pattern shown on the left hand side of Figure 3.2 is a very common example of such a color-filter. The values of the missing color bands at every pixel are often synthesized using some form of interpolation from neighboring pixel values. This process is known as color *demosaicing*.

Numerous demosaicing methods have been proposed through the years to solve this under-determined problem, and in this section we review some of the more popular ones. Of course, one can estimate the unknown pixel values by linear interpolation of the known ones in each color band independently. This approach will ignore some important information about the correlation between the color bands and will result in serious color artifacts. Note that with the Bayer pattern, the Red and Blue channels are down-sampled two times more than the Green channel. It is reasonable to assume that the independent interpolation of the Green band will result in a more reliable reconstruction than the Red or Blue bands. This property, combined

[2] This is the scenario for the more expensive 3-CCD cameras.

[3] This is the scenario for cheaper 1-CCD cameras.

with the assumption that the $\frac{Red}{Green}$ and $\frac{Blue}{Green}$ ratios are similar for the neighboring pixels, makes the basics of the smooth hue transition method first discussed in [52].

Note that there is a negligible correlation between the values of neighboring pixels located on the different sides of an edge. Therefore, although the smooth hue transition assumption is logical for smooth regions of the reconstructed image, it is not useful in the high-frequency (edge) areas. Considering this fact, gradient-based methods, first addressed in [3], do not preform interpolation across the edges of an image. This non-iterative method uses the second derivative of the Red and Blue channels to estimate the edge direction in the Green channel. Later, the Green channel is used to compute the missing values in the Red and Blue channels.

A variation of this method was later proposed in [53], where the second derivative of the Green channel and the first derivative of the Red (or Blue) channels are used to estimate the edge direction in the Green channel. The smooth hue and gradient based methods were later combined in [2]. In this iterative method, the smooth hue interpolation is done with respect to the local gradients computed in eight directions about a pixel of interest. A second stage using anisotropic inverse diffusion will further enhance the quality of the reconstructed image. This two step approach of interpolation followed by an enhancement step has been used in many other publications. In [54], spatial and spectral correlations among neighboring pixels are exploited to define the interpolation step, while adaptive median filtering is used as the enhancement step. A different iterative implementation of the median filter is used as the enhancement step of the method described in [55], that take advantage of a homogeneity assumption in the neighboring pixels.

Iterative MAP methods form another important category of demosaicing methods. A MAP algorithm with a smooth chrominance prior is discussed in [56]. The smooth chrominance prior is also used in [57], where the original image is first transformed to YIQ representation[4]. The chrominance interpolation is preformed using isotropic smoothing. The luminance interpolation is done using edge directions computed in a steerable wavelet pyramidal structure.

[4] YIQ is the standard color representation used in broadcast television (NTSC systems) [58].

Other examples of popular demosaicing methods available in published literature are [59], [60], [61], [62], [63], [64], and [65]. Almost all of the proposed demosaicing methods are based on one or more of these following assumptions:

1. In the measured image with the mosaic pattern, there are more green sensors with regular pattern of distribution than blue or red ones (in the case of Bayer CFA there are twice as many greens than red or blue pixels and each is surrounded by 4 green pixels).

2. Most algorithms assume a Bayer CFA pattern, for which each red, green and blue pixel is a neighbor to pixels of different color bands.

3. For each pixel, one and only one color band value is available.

4. The color pattern of available pixels does not change through the measured image.

5. The human eye is more sensitive to the details in the luminance component of the image than the details in chrominance component [57].

6. The human eye is more sensitive to chromatic changes in the low spatial frequency region than the luminance change [62].

7. Interpolation should be preformed along and not across the edges.

8. Different color bands are correlated with each other.

9. Edges should align between color channels.

Note that even the most popular and sophisticated demosaicing methods will fail to produce satisfactory results when severe aliasing is present in the color-filtered image. Such severe aliasing happens in cheap commercial still or video digital cameras, with small number of CCD pixels. The color artifacts worsen as the number of CCD pixels decreases. The following example shows this effect.

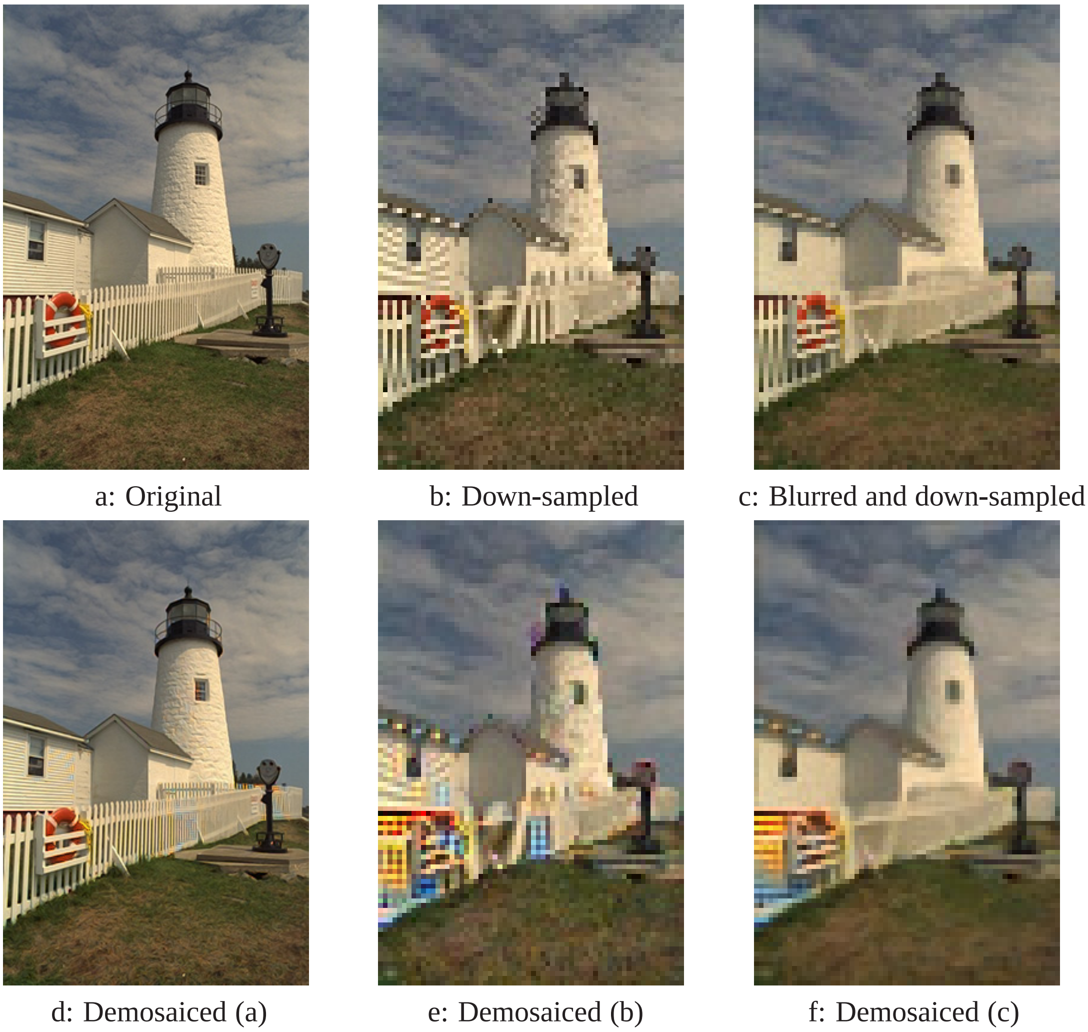
Figure 3.1.a shows a high-resolution image captured by a 3-CCD camera. If for capturing this image, instead of a 3-CCD camera a 1-CCD camera with the same number of CCD pixels was used, the inevitable mosaicing process will result in color artifacts. Figure 3.1.d shows the result of applying the demosaicing method of [2] with some negligible color-artifacts on the edges.

Note that many commercial digital video cameras can only be used in lower spatial resolution modes while working at higher frame rates. Figure 3.1.b shows a same scene from a 3-CCD camera with a down-sampling factor of 4 and Figure 3.1.e shows the demosaiced image of it after color-filtering. Note that the color artifacts in this image are much more evident than Figure 3.1.d. These color artifacts may be reduced by low-pass filtering the input data before color-filtering. Figure 3.1.c shows a factor of four down-sampled version of Figure 3.1.a, which is blurred with a symmetric Gaussian low-pass filter of size 4×4 with standard deviation equal to one, before down-sampling. The demosaiced image shown in Figure 3.1.f has less color artifacts than Figure 3.1.e, however it has lost some high-frequency details.

The poor quality of single-frame demosaiced images stimulates us to search for multi-frame demosaicing methods, where information from several low-quality images are fused together to produce high-quality demosaiced images.

3.2.3 Merging super-resolution and demosaicing into one process

Referring to the mosaic effects, the geometry of the single-frame and multi-frame demosaicing problems are fundamentally different, making it impossible to simply cross-apply traditional demosaicing algorithms to the multi-frame situation. To better understand the multi-frame demosaicing problem, we offer an example for the case of translational motion. Suppose that a set of color-filtered low-resolution images is available (images on the left in Figure 3.2). We use the two step process explained in Section 3.4 to fuse these images. The Shift-and-Add image on the right side of Figure 3.2 illustrates the pattern of sensor measurements in the high-resolution image grid. In such situations, the sampling pattern is quite arbitrary depending on the relative motion of the low-resolution images. This necessitates different demosaicing algorithms than those designed for the original Bayer pattern.


a: Original

b: Down-sampled

c: Blurred and down-sampled

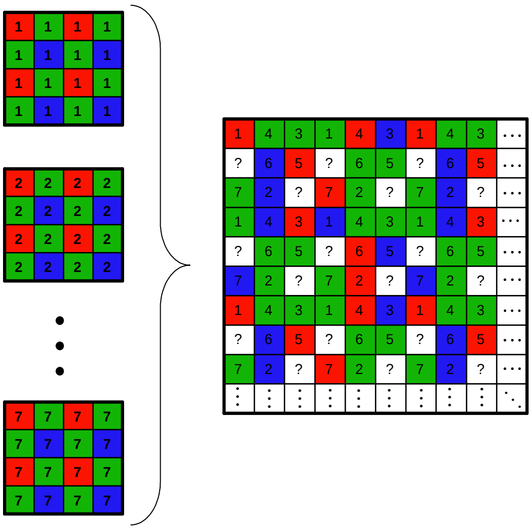
d: Demosaiced (a)

e: Demosaiced (b)

f: Demosaiced (c)

Figure 3.1: A high-resolution image (a) captured by a 3-CCD camera is down-sampled by a factor of four (b). In (c) the image in (a) is blurred by a Gaussian kernel before down-sampling by a factor of 4. The images in (a), (b), and (c) are color-filtered and then demosaiced by the method of [2]. The results are shown in (d), (e), (f), respectively.

Figure 3.2 shows that treating the green channel differently than the red or blue channels, as done in many single-frame demosaicing methods before, is not useful for the multi-frame case. While globally there are more green pixels than blue or red pixels, locally, any pixel may be surrounded by only red or blue colors. So, there is no general preference for one color band over the others (the first and second assumptions in Section 3.2.2 are not true for the



1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

2	2	2	2
2	2	2	2
2	2	2	2
2	2	2	2

7	7	7	7
7	7	7	7
7	7	7	7
7	7	7	7

1	4	3	1	4	3	1	4	3	⋯
?	6	5	?	6	5	?	6	5	⋯
7	2	?	7	2	?	7	2	?	⋯
1	4	3	1	4	3	1	4	3	⋯
?	6	5	?	6	5	?	6	5	⋯
7	2	?	7	2	?	7	2	?	⋯
1	4	3	1	4	3	1	4	3	⋯
?	6	5	?	6	5	?	6	5	⋯
7	2	?	7	2	?	7	2	?	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Figure 3.2: Fusion of 7 Bayer pattern low-resolution images with relative translational motion (the figures in the left side of the accolade) results in a high-resolution image ($\widehat{Z}$) that does not follow Bayer pattern (the figure in the right side of the accolade). The symbol "?" represents the high-resolution pixel values that were undetermined (as a result of insufficient low-resolution frames) after the Shift-and-Add step (Shift-and-Add method is extensively discussed in Chapter 2).

multi-frame case).

Another assumption, the availability of one and only one color band value for each pixel, is also not correct in the multi-frame case. In the under-determined cases, there are not enough measurements to fill the high-resolution grid. The symbol "?" in Figure 3.2 represents such pixels. On the other hand, in the over-determined cases, for some pixels, there may in fact be more than one color value available.

The fourth assumption in the existing demosaicing literature described earlier is not true because the field of view (FOV) of real world low-resolution images changes from one frame to the other, so the center and the border patterns of red, green, and blue pixels differ in the resulting high-resolution image.

3.3 Mathematical Model and Solution Outline

3.3.1 Mathematical Model of the Imaging System

In the previous chapter, we studied several distorting processes such as warping, blurring, and additive noise that affect the quality of images acquired by commercial digital cameras. These effects were illustrated in Figure 2.1 and mathematically modeled in (2.4). In this chapter, we generalize this imaging system model to also consider the color-filtering effects as illustrated in Figure 3.3. In this model, a real-world scene is seen to be warped at the camera lens because of the relative motion between the scene and camera. The optical lens and aperture result in the blurring of this warped image which is then sub-sampled and color-filtered at the CCD. The additive readout noise at the CCD will further degrade the quality of captured images.

We represent this approximated forward model by the following equation

$$\begin{aligned} \underline{Y}_i(k) &= D_i(k)H(k)F(k)\underline{X}_i + \underline{V}_i(k) \\ &= M_i(k)\underline{X}_i + \underline{V}_i(k) \qquad k = 1,\ldots,N \qquad i = R,G,B \quad , \end{aligned} \tag{3.1}$$

which can be also expressed as:

$$\underline{Y} = M\underline{X}+\underline{V}, \qquad \underline{Y} = \begin{bmatrix} \underline{Y}_R(1) \\ \underline{Y}_G(1) \\ \underline{Y}_B(1) \\ \underline{Y}_R(2) \\ \vdots \\ \underline{Y}_B(N) \end{bmatrix}, \underline{V} = \begin{bmatrix} \underline{V}_R(1) \\ \underline{V}_G(1) \\ \underline{V}_B(1) \\ \underline{V}_R(2) \\ \vdots \\ \underline{V}_B(N) \end{bmatrix}, M = \begin{bmatrix} \underline{M}_R(1) \\ \underline{M}_G(1) \\ \underline{M}_B(1) \\ \underline{M}_R(2) \\ \vdots \\ \underline{M}_B(N) \end{bmatrix}, \underline{X} = \begin{bmatrix} \underline{X}_R \\ \underline{X}_G \\ \underline{X}_B \end{bmatrix}. \tag{3.2}$$

The vectors $\underline{X}_i$ and $\underline{Y}_i(k)$ are representing the i^{th} band (R, G, or B) of the high-resolution color frame and the k^{th} low-resolution frame after lexicographic ordering, respectively. Matrix $F(k)$ is the geometric motion operator between the high-resolution and low-resolution frames. The camera's point spread function (PSF) is modeled by the blur matrix $H(k)$. The matrix $D_i(k)$

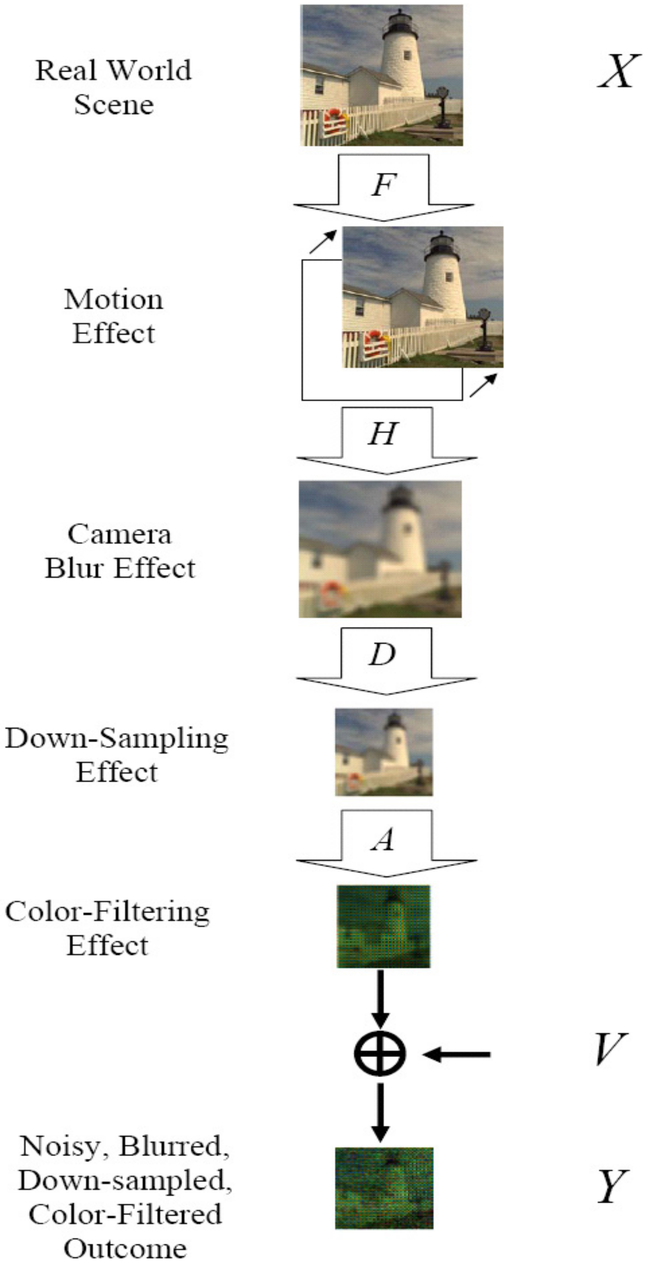


Figure 3.3: Block diagram representing the image formation model considered in this chapter, where X is the intensity distribution of the scene, V is the additive noise, and Y is the resulting color-filtered low-quality image. The operators F, H, D, and A are representatives of the warping, blurring, down-sampling, and color-filtering processes, respectively.

represents the down-sampling operator, which includes both the color-filtering and CCD down-sampling operations[5]. Geometric motion, blur, and down-sampling operators are covered by the operator $M_i(k)$, which we call the system matrix. The vector $\underline{V}_i(k)$ is the system noise and N is the number of available low-resolution frames.

The high-resolution color image ($\underline{X}$) is of size $[12r^2Q_1Q_2 \times 1]$), where r is the resolution enhancement factor. The size of the vectors $\underline{V}_G(k)$ and $\underline{Y}_G(k)$ is $[2Q_1Q_2 \times 1]$ and vectors $\underline{V}_R(k)$, $\underline{Y}_R(k)$, $\underline{V}_B(k)$, and $\underline{Y}_B(k)$ are of size $[Q_1Q_2 \times 1]$. The geometric motion and blur matrices are of size $[4r^2Q_1Q_2 \times 4r^2Q_1Q_2]$. The down-sampling and system matrices are of size $[2Q_1Q_2 \times 4r^2Q_1Q_2]$ for the Green band, and of size $[Q_1Q_2 \times 4r^2Q_1Q_2]$ for the Red and Blue bands[6].

Considered separately, super-resolution and demosaicing models are special cases of the general model presented above. In particular, in the super-resolution literature the effect of color-filtering is usually ignored [4, 11, 47] and therefore the model is simplified to

$$\underline{Y}(k) = D(k)H(k)F(k)\underline{X} + \underline{V}(k) \qquad k = 1, \ldots, N \quad . \tag{3.3}$$

In this model (as explained in the previous chapter) the low-resolution images $\underline{Y}(k)$ and the high-resolution image $\underline{X}$ are assumed to be monochromatic. On the other hand, in the demosaicing literature only single frame reconstruction of color images is considered, resulting in the simplified model

$$\underline{Y}_i = D_i\underline{X}_i + \underline{V}_i \qquad i = R, G, B \quad . \tag{3.4}$$

As such, the classical approach to the multi-frame reconstruction of color images has been a two-step process. The first step is to solve (3.4) for each image (demosaicing step) and the second step is to use the model in (3.3) to fuse the low-resolution images resulting from the first step, reconstructing the color high-resolution image (usually each R, G , or B bands is processed individually). Figure 3.4 illustrates the block diagram representation of this method.

[5] It is convenient to think of $D_i(k) = A_i(k)D(k)$, where $D(k)$ models the down-sampling effect of the CCD and $A_i(k)$ models the color-filter effect [66].

[6] Note that color super-resolution by itself is a special case of this model, where vectors $\underline{V}_i(k)$ and $\underline{Y}_i(k)$ are of size $[4Q_1Q_2 \times 1]$ and matrices $M_i(k)$ and $D_i(k)$ are of size $[4Q_1Q_2 \times 4r^2Q_1Q_2]$ for any color band.

Of course, this two step method is a suboptimal approach to solving the overall problem. In Section 3.4, we propose a Maximum A-Posteriori (MAP) estimation approach to directly solve (3.2). Figure 3.5 illustrates the block diagram representation of our proposed method.

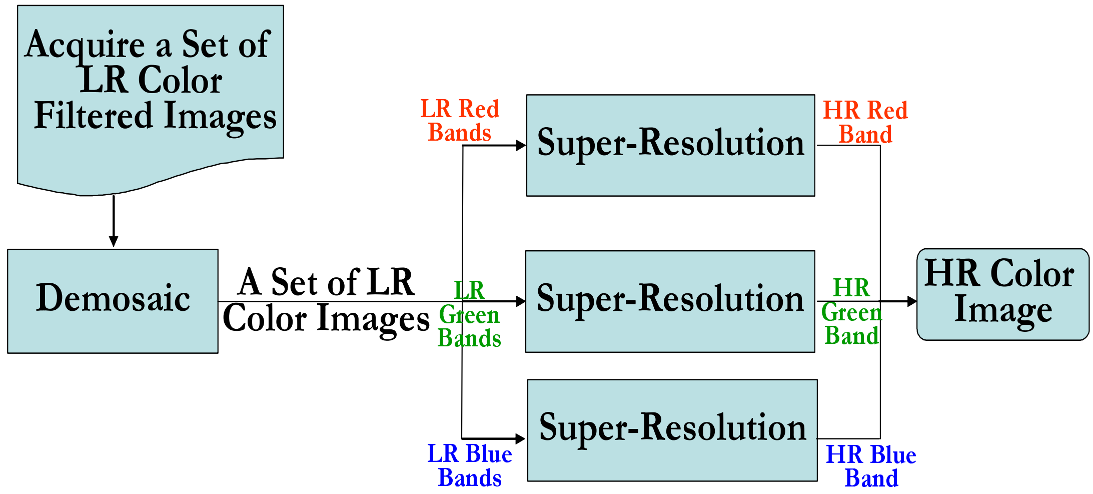


Figure 3.4: Block diagram representing the classical approach to the multi-frame reconstruction of color images.

3.4 Multi-Frame Demosaicing

In Section 3.2.3 we indicated how the multi-frame demosaicing is fundamentally different than single-frame demosaicing. In this section, we propose a computationally efficient MAP estimation method to fuse *and* demosaic a set of low-resolution frames (which may have been color-filtered by any CFA) resulting in a color image with higher spatial resolution and reduced color artifacts. Our MAP based cost function consists of the following terms, briefly motivated in the previous section:

1. A penalty term to enforce similarities between the raw data and the high-resolution estimate (Data Fidelity Penalty Term).

2. A penalty term to encourage sharp edges in the luminance component of the high-resolution

image (Spatial Luminance Penalty Term).

3. A penalty term to encourage smoothness in the chrominance component of the high-resolution image (Spatial Chrominance Penalty Term).

4. A penalty term to encourage homogeneity of the edge location and orientation in different color bands (Inter-Color Dependencies Penalty Term).

Each of these penalty terms will be discussed in more detail in the following subsections.

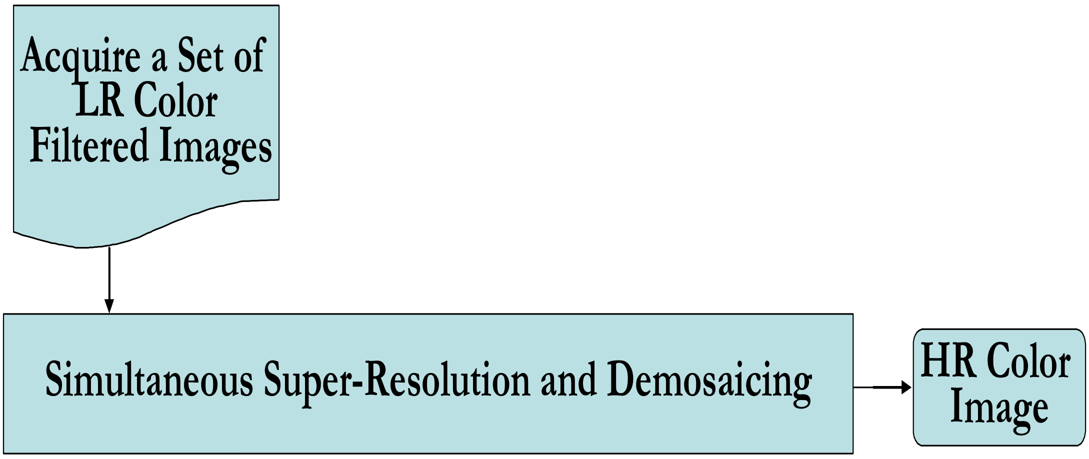

Acquire a Set of LR Color Filtered Images
Simultaneous Super-Resolution and Demosaicing
HR Color Image

Figure 3.5: Block diagram representing the proposed direct approach to the multi-frame reconstruction of color images.

3.4.1 Data Fidelity Penalty Term

This term measures the similarity between the resulting high-resolution image and the original low-resolution images. As explained in Section 2.2 and [4], L_1 norm minimization of the error term results in robust reconstruction of the high-resolution image in the presence of uncertainties such as motion error. Considering the general motion and blur model of (3.2), the (multi spectral) data fidelity penalty term is defined as:

$$J_0(\underline{X}) = \sum_{i=R,G,B} \sum_{k=1}^{N} \|D_i(k)H(k)F(k)\underline{X}_i - \underline{Y}_i(k)\|_1. \tag{3.5}$$

Note that the above penalty function is applicable for general models of data, blur and motion. However, in this chapter we only treat the simpler case of common space invariant PSF and translational motion. This could, for example, correspond to a vibrating camera acquiring a sequence of images from a static scene.

For this purpose, we use the two step method of Section 2.2.5 to represent the data fidelity penalty term, which is easier to interpret and has a faster implementation potential [4]. This simplified data fidelity penalty term is defined as

$$J_0(\underline{X}) = \sum_{i=R,G,B} \|\Phi_i \left(H\widehat{\underline{X}}_i - \widehat{\underline{Z}}_i \right) \|_1 \ , \tag{3.6}$$

where $\widehat{\underline{Z}}_R$, $\widehat{\underline{Z}}_G$, and $\widehat{\underline{Z}}_B$ are the three color channels of the color Shift-and-Add image, $\widehat{\underline{Z}}$. The matrix Φ_i ($i = R, G, B$), is a diagonal matrix with diagonal values equal to the square root of the number of measurements that contributed to make each element of $\widehat{\underline{Z}}_i$ (in the square case is the identity matrix). So, the undefined pixels of $\widehat{\underline{Z}}_B$ have no effect on the high-resolution estimate. On the other hand, those pixels of $\widehat{\underline{Z}}_B$ which have been produced from numerous measurements, have a stronger effect in the estimation of the high-resolution frame. The vectors $\widehat{\underline{X}}_R$, $\widehat{\underline{X}}_G$, and $\widehat{\underline{X}}_B$ are the three color components of the reconstructed high-resolution image $\widehat{X}$. Figure 3.6 illustrates the block diagram representation of this fast two-step method.

3.4.2 Spatial Luminance Penalty Term

The human eye is more sensitive to the details in the luminance component of an image than the details in the chrominance components [57]. Therefore, it is important that the edges in the luminance component of the reconstructed high-resolution image look sharp. As explained in Section 2.2.3, applying BTV regularization to the luminance component will result in this desired property [4]. The luminance image can be calculated as the weighted sum $\underline{X}_L = 0.299\underline{X}_R + 0.597\underline{X}_G + 0.114\underline{X}_B$ as explained in [58]. The luminance regularization term is then defined as

$$J_1(\underline{X}) = \sum_{l,m=-P}^{P} \alpha^{|m|+|l|} \|\underline{X}_L - S_x^l S_y^m \underline{X}_L\|_1. \tag{3.7}$$

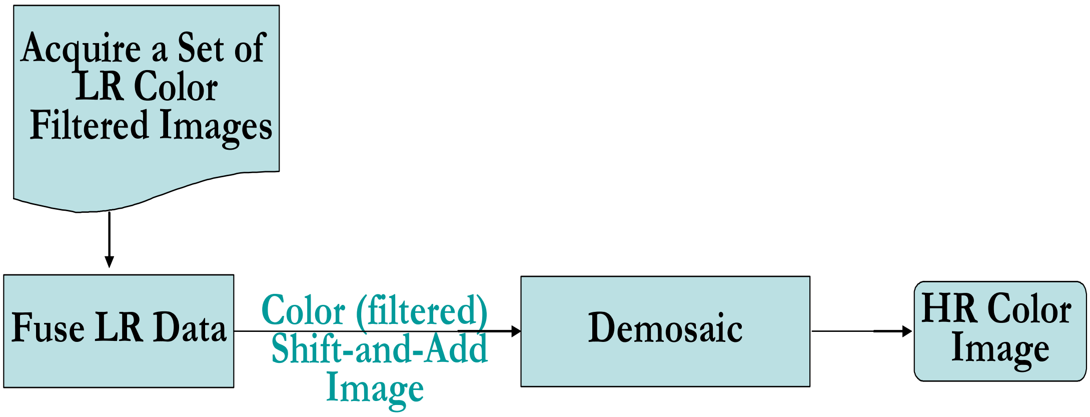

Acquire a Set of LR Color Filtered Images
Fuse LR Data
Color (filtered) Shift-and-Add Image
Demosaic
HR Color Image

Figure 3.6: Block diagram representing the proposed fast two-step approach (3.6) to the multi-frame reconstruction of color images, applicable to the case of common space invariant PSF and translational motion.

3.4.3 Spatial Chrominance Penalty Term

Spatial regularization is required also for the chrominance layers. However, since the human visual system is less sensitive to the resolution of these bands, we can use a simpler regularization, based on the L_2 norm (similar to the Tikhonov regularization discussed in Section 2.2.3)

$$J_2(\underline{X}) = \|\Lambda \underline{X}_{C_1}\|_2^2 + \|\Lambda \underline{X}_{C_2}\|_2^2 \ , \tag{3.8}$$

where the images $\underline{X}_{C_1}$ and $\underline{X}_{C_2}$ are the I and Q layers in the YIQ color representation[7], and Λ is a high-pass operator such as derivative, Laplacian, or even identity matrix.

3.4.4 Inter-Color Dependencies Penalty Term

This term penalizes the mismatch between locations or orientations of edges across the color bands. Following [56], minimizing the vector product norm of any two adjacent color pixels forces different bands to have similar edge location and orientation. The squared-norm of the vector (outer) product of $\underline{U} : [u_r, u_g, u_b]^T$ and $\underline{W} : [w_r, w_g, w_b]^T$, which represent the

[7] The Y layer ($\underline{X}_L$) is treated in (3.7).

color values of two adjacent pixels, is defined as

$$\begin{aligned}\| \underline{U} \times \underline{W} \|_2^2 &= \|\underline{U}\|_2^2 \|\underline{W}\|_2^2 \sin^2(\Theta) = \|\vec{i}(u_g w_b - u_b w_g)\|_2^2 \\ &+ \|\vec{j}(u_b w_r - u_r w_b)\|_2^2 + \|\vec{k}(u_r w_g - u_g w_r)\|_2^2, \qquad (3.9)\end{aligned}$$

where Θ is the angle between these two vectors, and $\vec{i}, \vec{j}, \vec{k}$ are the principal direction vectors in 3-D. As the data fidelity penalty term will restrict the values of $\|\underline{U}\|$ and $\|\underline{W}\|$, minimization of $\|\underline{U} \times \underline{W}\|_2^2$ will minimize $\sin(\Theta)$, and consequently Θ itself, where a small value of Θ is an indicator of similar orientation. Based on the theoretical justifications of [67], the authors of [56] suggest a pixelwise inter-color dependencies cost function to be minimized. This term has the vector outer product norm of all pairs of neighboring pixels, which is solved by the finite element method.

With some modifications to what was proposed in [56], our inter-color dependencies penalty term is a differentiable cost function

$$\begin{aligned}J_3(\underline{X}) &= \sum_{l,m=-1}^{1} \Big[\|\underline{X}_G \odot S_x^l S_y^m \underline{X}_B - \underline{X}_B \odot S_x^l S_y^m \underline{X}_G\|_2^2 + \\ &\|\underline{X}_B \odot S_x^l S_y^m \underline{X}_R - \underline{X}_R \odot S_x^l S_y^m \underline{X}_B\|_2^2 + \|\underline{X}_R \odot S_x^l S_y^m \underline{X}_G - \underline{X}_G \odot S_x^l S_y^m \underline{X}_R\|_2^2 \Big], \qquad (3.10)\end{aligned}$$

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

3.4.5 Overall Cost Function

The overall cost function is the combination of the cost functions described in the previous subsections:

$$\widehat{\underline{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[J_0(\underline{X}) + \lambda' J_1(\underline{X}) + \lambda'' J_2(\underline{X}) + \lambda''' J_3(\underline{X}) \right] \ . \qquad (3.11)$$

A version of steepest descent optimization may be applied to minimize this cost function. In the first step, the derivative of (3.11) with respect to one of the color bands is calculated, assuming the other two color bands are fixed. In the next steps, the derivative will be computed with respect to the other color channels. For example the derivative with respect to the Green band ($\underline{X}_G$) is calculated as follows

$$\begin{aligned}\nabla\underline{\widehat{X}}_G^n &= H^T\Phi_G^T\text{sign}(\Phi_G H\underline{\widehat{X}}_G^n - \Phi_G\underline{\widehat{Z}}_G) + \\ &\lambda' \sum_{l,m=-P}^{P} \alpha^{|m|+|l|} \times 0.5870 \times [I - S_y^{-m}S_x^{-l}]\text{sign}\Big(0.2989(\underline{X}_R^n - S_x^l S_y^m \underline{X}_R^n) + \\ &0.5870(\underline{X}_G^n - S_x^l S_y^m \underline{X}_G^n) + 0.1140(\underline{X}_B^n - S_x^l S_y^m \underline{X}_B^n)\Big) + \\ &\lambda'' \sum_{l,m=-1}^{1} \Big[2(\mathbf{X_B^{l,m}} - S_x^{-l}S_y^{-m}\mathbf{X_B})(\mathbf{X_B^{l,m}}\underline{X}_G - \mathbf{X_B}S_x^l S_y^m \underline{X}_G) + \\ &2(\mathbf{X_R^{l,m}} - S_x^{-l}S_y^{-m}\mathbf{X_R})(\mathbf{X_R^{l,m}}\underline{X}_G - \mathbf{X_R}S_x^l S_y^m \underline{X}_G)\Big] + \\ &\lambda''' \Lambda^T\Lambda(-0.1536 \times \underline{X}_R + 0.2851 \times \underline{X}_G - 0.1316 \times \underline{X}_B), \qquad (3.12)\end{aligned}$$

where S_x^{-l} and S_y^{-m} define the transposes of matrices S_x^l and S_y^m, respectively, and have a shifting effect in the opposite directions of S_x^l and S_y^m. The notation $\mathbf{X_R}$, and $\mathbf{X_B}$ stands for the diagonal matrix representations of the Red and Blue bands and $\mathbf{X_R^{l,m}}$ and $\mathbf{X_B^{l,m}}$ are the diagonal representations of these matrices shifted by l and m pixels in the horizontal and vertical directions, respectively. The calculation of the inter-color dependencies term derivative is explained in the Appendix E.

Similar to the grayscale super-resolution case, matrices H, Λ, Φ, D, S_x^l, and S_y^m and their transposes can be exactly interpreted as direct image operators such as blur, high-pass filtering, masking, down-sampling, and shift. Noting and implementing the effects of these matrices as a sequence of operators on the images directly spares us from explicitly constructing them as matrices. This property helps our method to be implemented in a fast and memory efficient way.

The gradient of the other channels will be computed in the same way, and the following steepest (coordinate) descent iterations is used to calculate the high-resolution image estimate iteratively.

$$\underline{\widehat{X}}_i^{n+1} = \underline{\widehat{X}}_i^n - \beta\nabla\underline{\widehat{X}}_i^n \qquad i = R, G, B \qquad , \qquad (3.13)$$

where the scalar β is the step size.

3.5 Related Methods

As mentioned earlier, there has been very little work on the problem we have posed here. One related paper is the work of Zomet and Peleg [68], who have recently proposed a novel method for combining the information from multiple sensors, which can also be used for demosaicing purposes. Although their method has produced successful results for the single frame demosaicing problem, it is not specifically posed or directed towards solving the multi-frame demosaicing problem, and no multi-frame demosaicing case experiment is given.

The method of [68] is based on the assumption of affine relation between the *intensities* of different sensors in a local neighborhood. To estimate the Red channel, first, affine relations that project Green and Blue channels to the Red channel are computed. In the second stage, a super-resolution algorithm (e.g. the method of [25]) is applied on the available low-resolution images in the Red channel (i.e. the original CFA data of the Red channel plus the projected Green and Blue channels) to estimate the high-resolution Red channel image. A similar procedure estimates the high-resolution Green and Blue channel images. As an affine model is not always valid for all sensors or image sets, an affine model validity test is utilized in [68]. In the case that the affine model is not valid for some pixels, those projected pixels are simply ignored.

The method of [68] is strongly dependent on the validity of the affine model, which is not confirmed for the multi-frame case with inaccurate registration artifacts. Besides, the original CFA low-resolution image of a channel (raw data) and the less reliable projected low-resolution images of other channels are equally weighted to construct the missing values, and this does not appear to be an optimal solution.

In contrast to their method, our proposed technique exploits the correlation of the information in different channels explicitly to guarantee similar edge position and orientation in different color bands. Our proposed method also exploits the difference in sensitivity of the human eye to the frequency content and outliers in the luminance and chrominance components of the image.

In parallel to our work, Gotoh and Okotumi [69] are proposing another MAP estimation method for solving the same joint demosaicing/super-resolution problem. While their algorithm and ours share much in common, there are fundamental differences between the two in terms of robustness to model errors, and prior used. Model errors, such as choice of blur or motion estimation errors, are treated favorably by our algorithm due to the L_1 norm employed in the likelihood fidelity term. By contrast, in [69], an L_2-norm data fusion term is used, which is not robust to such errors. In [4] it is shown how this difference in norm can become crucial in obtaining better results in the presence of model mismatches.

As to the choice of prior, ours is built of several pieces, giving an overall edge preserved outcome, smoothed chrominance layers, and forced edge and orientation alignment between color layers. To the contrary, [69] utilizes an anisotropic Tikhonov (L_2 norm) method of regularizing.

3.6 Experiments

Experiments on synthetic and real data sets are presented in this section. In the first experiment, following the model of (3.2), we created a sequence of low-resolution frames from an original high-resolution image (Figure 3.7(a)), which is a color image with full RGB values. First we shifted this high-resolution image by one pixel in the vertical direction. Then to simulate the effect of camera PSF, each color band of this shifted image was convolved with a symmetric Gaussian low-pass filter of size 5×5 with standard deviation equal to one. The resulting image was subsampled by the factor of 4 in each direction. The same process with different motion vectors (shifts) in vertical and horizontal directions was used to produce 10 low-resolution images from the original scene. The horizontal shift between the low-resolution images was varied between 0 to .75 pixels in the low-resolution grid (0 to 3 pixels in the high-resolution grid). The vertical shift between the low-resolution images varied between 0 to .5 pixels in the low-resolution grid (0 to 2 pixels in the high-resolution grid). To simulate the errors in motion estimation, a bias equal to half a pixel shift in the low-resolution grid was intentionally added to the known motion vector of one of the low-resolution frames. We added Gaussian noise to the resulting low-resolution frames to achieve SNR equal to 30dB. Then each low-resolution color image was subsampled by the Bayer filter.

One of these Bayer filtered low-resolution images is reconstructed by the demosaicing method of [3] and shown in Figure 3.7(b). The above method is implemented on Kodak DCS-200 digital cameras [70], so each low-resolution image may be thought of as one picture taken with this camera brand. Figure 3.7(c) shows the result of using the more sophisticated demosaicing method [8] of [2].

As the motion model for this experiment is translational and the blur kernel is space invariant, we can use the fast model of (3.12) to reconstruct the blurry image ($\widehat{\underline{Z}}$) on the high-resolution grid. The Shift-and-Add result of the demosaiced low-resolution frames after bilinear interpolation[9], before deblurring and demosaicing is shown in Figure 3.7(d). We used the result of the Shift-and-Add method as the initialization of the iterative multi-frame demosaicing methods. We used the original set of frames (raw data) to reconstruct a high-resolution image with reduced color artifacts. Figures 3.8(a), 3.8(b), and 3.8(c) show the effect of the individual implementation of each regularization term (luminance, chrominance, and inter-color dependencies), described in Section 3.4.

We applied the method of [2] to demosaic each of these 10 low-resolution frames individually, and then applied the robust super-resolution method of [4] (Chapter 2) on each resulting color channel. The result of this method is shown in Figure 3.8(d). We also applied

[8] We thank Prof. Ron Kimmel of the Technion for providing us with the code that implements the method in [2].

[9] Interpolation is needed as this experiment is an under-determined problem, where some pixel values are missing.

the robust super-resolution method of [4] on the raw (Bayer filtered) data (before demosaicing)[10]. The result of this method is shown in Figure 3.9(a). To study the effectiveness of each regularization term, we paired (inter-color dependencies-luminance, inter-color dependencies-chrominance, and luminance-chrominance) regularization terms for which the results are shown in Figures 3.9(b), 3.9(c), and 3.9(d) ,respectively. Finally, Figure 3.10(a) shows the result of the implementation of (3.11) with all terms. The parameters used for this example are as follows: $\beta = 0.002$, $\alpha = 0.9$, $\lambda^{'} = 0.01$, $\lambda^{''} = 150$, $\lambda^{'''} = 1$.

It is clear that the resulting image (Figure 3.10(a)) has a better quality than the low-resolution input frames or other reconstruction methods. Quantitative measurements confirm this visual comparison. We used PSNR [11] and S-CIELAB [12] measures to compare the performance of each of these methods. Table 3.1 compares these values in which the proposed method has the lowest S-CIELAB error and the highest PSNR values (and also the best visual quality specially in the red lifesaver section of the image).

In the second experiment, we used 30 compressed images captured from a commercial webcam (PYRO-1394). Figure 3.11(a) shows one of these low-resolution images (a selected region of this image is zoomed in Figure 3.11(e) for closer examination). Note that the compression (blocking) and color artifacts are quite apparent in these images. This set of frames was already demosaiced, and no information was available about the original sensor values, which makes the color enhancement task more difficult. This example may be also considered as a multi-frame color super-resolution case. The (unknown) camera PSF was assumed to be a

[10] To apply the monochromatic SR method of [4] on this color-filtered sequence, we treated each color band separately. To consider the color-filtering operation, we substituted matrix A in Equation (23) of [4] with matrix Φ in (3.6).

[11] The PSNR of two vectors $\underline{X}$ and $\underline{\widehat{X}}$ of size $[4r^2Q_1Q_2 \times 1]$ is defined as:

$$\text{PSNR}(\underline{X}, \underline{\widehat{X}}) = 10\log_{10}\left(\frac{255^2 \times 4r^2Q_1Q_2}{\|\underline{X} - \underline{\widehat{X}}\|_2^2}\right).$$

[12] The S-CIELAB measure is a perceptual color fidelity measure that measures how accurate the reproduction of a color is to the original when viewed by a human observer [71]. In our experiments, we used the code with default parameters used in the implementation of this measure available at http://white.stanford.edu/∼brian/scielab/scielab.html .

	Shift-and-Add	LR Demosaiced [2]+SR [4]	Only Lumin.	Only Orient.	Only Chromin. .
S-CIELAB	1.532×10^{11}	1.349×10^{11}	7.892×10^{10}	6.498×10^{10}	4.648×10^{10}
PSNR (dB)	17.17	19.12	17.74	20.10	20.35
	SR [4] on Raw Data	Lumin. +Orient.	Orient. +Chrom.	Lumin. +Chrom.	Full
S-CIELAB	5.456×10^{10}	4.543×10^{10}	4.382×10^{10}	3.548×10^{10}	3.365×10^{10}
PSNR (dB)	19.28	20.79	20.68	21.12	21.13

Table 3.1: The quantitative comparison of the performance of different demosaicing methods on the lighthouse sequence. The proposed method has the lowest S-CIELAB error and the highest PSNR value.

4×4 Gaussian kernel with standard deviation equal to one for each color band. As the relative motion between these images followed the translational model, we only needed to estimate the motion between the luminance components of these images [72]. We used the method described in [48] to compute the motion vectors.

The Shift-and-Add result (resolution enhancement factor of 4) is shown in Figure 3.11(b) (zoomed in Figure 3.11(f)). In Figure 3.11(c) (zoomed in Figure 3.11(g)) the method of [4] is used for increasing the resolution by a factor of 4 in each color band, independently. And finally the result of applying our method on this sequence is shown in Figure 3.11(d) (zoomed in Figure 3.11(h)), where color artifacts are significantly reduced. The parameters used for this example are as follows: $\beta = 0.004$, $\alpha = 0.9$, $\lambda^{'} = 0.25$, $\lambda^{''} = 500$, $\lambda^{'''} = 5$.

In the third experiment, we used 40 compressed images of a test pattern from a surveillance camera; courtesy of Adyoron Intelligent Systems Ltd., Tel Aviv, Israel. Figure 3.12(a) shows one of these low-resolution images (a selected region of this image is zoomed in Figure 3.13.a for closer examination). Note that the compression (blocking) and color artifacts are quite apparent in these images. This set of frames was also already demosaiced, and no information was available about the original sensor values. This example may be also considered as a multi-frame color super-resolution case. The (unknown) camera PSF was assumed to be a 6×6 Gaussian kernel with standard deviation equal to two for each color band.

We used the method described in [48] to compute the motion vectors. The Shift-and-Add result (resolution enhancement factor of 4) is shown in Figure 3.12(b) (zoomed in Figure 3.13(b)). In Figure 3.12(c) (zoomed in Figure 3.13(c)) the method of [4] is used for increasing the resolution by a factor of 4 in each color band, independently. And finally the result of applying the proposed method on this sequence is shown in Figure 3.12(d), (zoomed in Figure 3.13(d)), where color artifacts are significantly reduced. Moreover, comparing to the Figures 3.12(a)-(d), the compression errors have been removed more effectively in Figures 3.12(d). The parameters used for this example are as follows: $\beta = 0.004$, $\alpha = 0.9$, $\lambda^{'} = 0.25$, $\lambda^{''} = 500$, $\lambda^{'''} = 5$.

In the fourth, fifth, and sixth experiments (Girl, Bookcase, and Window sequences), we used 31 uncompressed, raw CFA images (30 frames for the Window sequence) from a video camera (based on Zoran 2MP CMOS Sensors using the coach chipset). We applied the method of [3] to demosaic each of these low-resolution frames, individually. Figure 3.14(a) (zoomed in Figure 3.15(a)) shows one of these images from the Girl sequence (corresponding image of the Bookcase sequence is shown in Figure 3.16(a) and the corresponding image of the Window sequence is shown in Figure 3.18(a)). The result of the more sophisticated demosaicing method of [2] for Girl sequence is shown in Figure 3.14(b) (zoomed in Figure 3.15(b)). Figure 3.16(b) shows the corresponding image for the Bookcase sequence and Figure 3.18(b) shows the corresponding image for the Window sequence.

To increase the spatial resolution by a factor of three, we applied the proposed multi-frame color super-resolution method on the demosaiced images of these two sequences. Figure 3.14(c) shows the high-resolution color super-resolution result from the low-resolution color images of Girl sequence demosaiced by the method of [3] (zoomed in Figure 3.15(c)). Figure 3.16(c) shows the corresponding image for the Bookcase sequence and Figure 3.18(c) shows the corresponding image for the Window sequence. Similarly, Figure 3.14(d) shows the result of resolution enhancement of the low-resolution color images from Girl sequence demosaiced by the method of [2] (zoomed in Figure 3.15(d)). Figure 3.16(d) shows the corresponding image for the Bookcase sequence and Figure 3.18(d) shows the corresponding image for the Window sequence.

Finally, we directly applied the proposed multi-frame demosaicing method on the raw CFA data to increase the spatial resolution by the same factor of three. Figure 3.14(e) shows the high-resolution result of multi-frame demosaicing of the low-resolution raw CFA images from Girl sequence without using the inter color dependence term [$J_3(\underline{X})$] (zoomed in Figure 3.15(e)). Figure 3.17(a) shows the corresponding image for the Bookcase sequence and Figure 3.18(e) shows the corresponding image for the Window sequence. Figure 3.14(f) shows the high-resolution result of applying the multi-frame demosaicing method using all proposed terms in (3.11) on the low-resolution raw CFA images from Girl sequence (zoomed in Figure 3.15(f)). Figure 3.17(b) shows the corresponding image for the Bookcase sequence and Figure 3.18(f) shows the corresponding image for the Window sequence.

These experiments show that single frame demosaicing methods such as [2] (which in effect implement de-aliasing filters) remove color artifacts at the expense of making the images more blurry. The proposed color super-resolution algorithm can retrieve some high frequency information and further remove the color artifacts. However, applying the proposed multi-frame demosaicing method directly on raw CFA data produces the sharpest results and effectively removes color artifacts. These experiments also show the importance of the inter-color dependence term which further removes color artifacts. The parameters used for the experiments on Girl, Bookcase, and Window sequences are as follows: $\beta = 0.002$, $\alpha = 0.9$, $\lambda^{'} = 0.1$, $\lambda^{''} = 250$, $\lambda^{'''} = 25$. The (unknown) camera PSF was assumed to be a tapered 5×5 disk PSF [13].

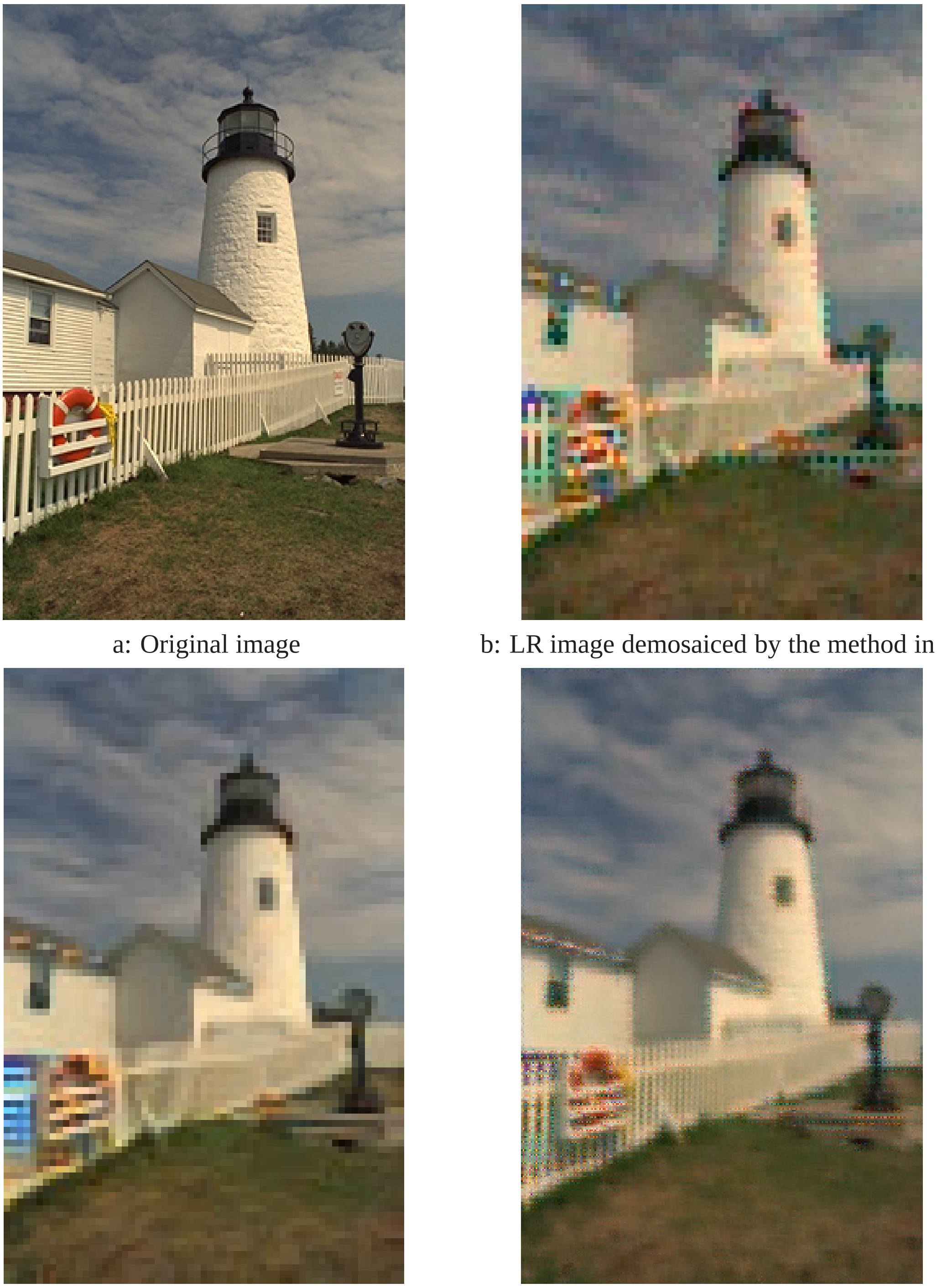
3.7 Summary and Discussion

In this chapter, based on the MAP estimation framework, we proposed a unified method of demosaicing and super-resolution, which increases the spatial resolution and reduces

[13] MATLAB command fspecial(‘disk’,2) creates such blurring kernel.

the color artifacts of a set of low-quality color images. Using the L_1 norm for the data error term makes our method robust to errors in data and modeling. Bilateral regularization of the luminance term results in sharp reconstruction of edges, and the chrominance and inter-color dependencies cost functions remove the color artifacts from the high-resolution estimate. All matrix-vector operations in the proposed method are implemented as simple image operators. As these operations are locally performed on pixel values on the high-resolution grid, parallel processing may also be used to further increase the computational efficiency. The computational complexity of this method is on the order of the computational complexity of the popular iterative super-resolution algorithms, such as [11]. Namely, it is linear in the number of pixels.

The inter-color dependencies term (3.10) results in the non-convexity of the overall penalty function. Therefore, the steepest decent optimization of (3.11) may reach a local rather than the global minimum of the overall function. The non-convexity does not impose a serious problem if a reasonable initial guess is used for the steepest decent method, as many experiments showed effective multi-frame demosaicing results. In our experiments we noted that a good initial guess is the Shift-and-Add result of the individually demosaiced low-resolution images.



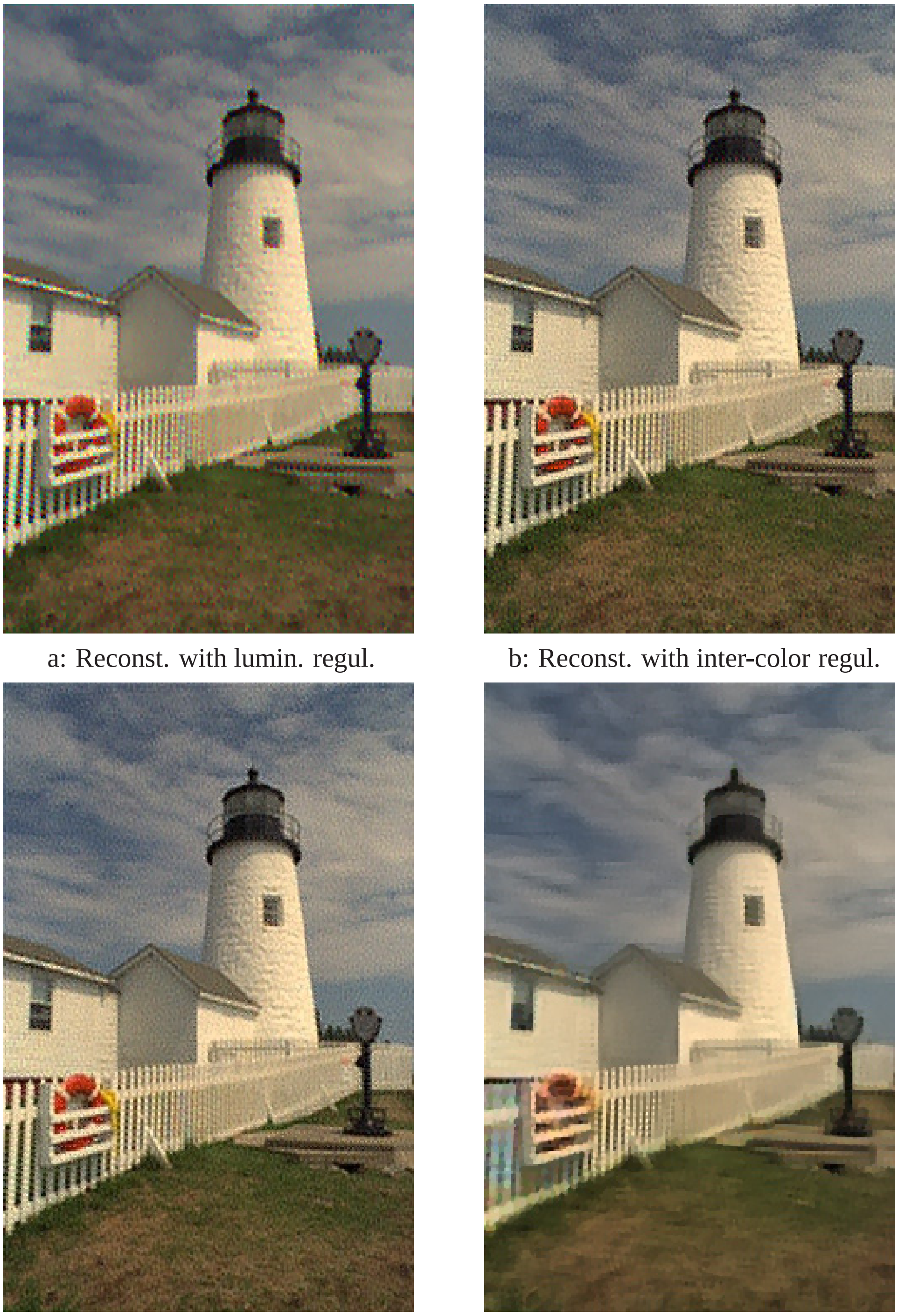
a: Original image

b: LR image demosaiced by the method in [3]

c: LR image demosaiced by the method in [2]

d: Shift-and-Add image.

Figure 3.7: A high-resolution image (a) of size $[384 \times 256 \times 3]$ is passed through our model of camera to produce a set of low-resolution images. One of these low-resolution images is demosaiced by the method in [3] (b) (low-resolution image of size $[96 \times 64 \times 3]$). The same image is demosaiced by the method in [2] (c). Shift-and-Add on the 10 input low-resolution images is shown in (d) (of size $[384 \times 256 \times 3]$).



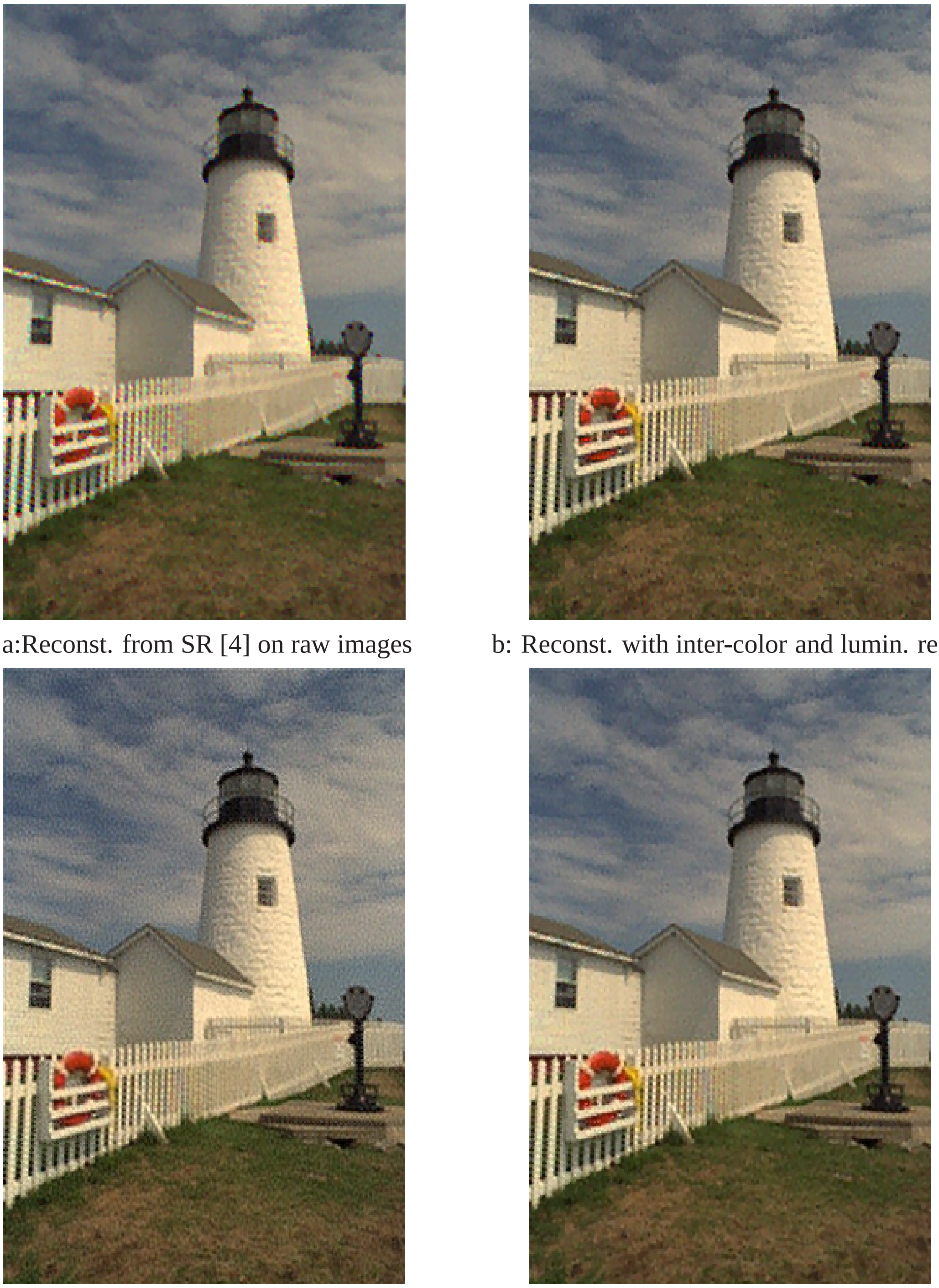
a: Reconst. with lumin. regul.

b: Reconst. with inter-color regul.

c: Reconst. with chromin. regul.

d: Reconst. from LR demosaiced [2]+SR [4]

Figure 3.8: Multi-frame demosaicing of this set of low-resolution frames with the help of only luminance, inter-color dependencies or chrominance regularization terms is shown in (a), (b), and (c), respectively. The result of applying the super-resolution method of [4] on the low-resolution frames each demosaiced by the method [2] is shown in (d).



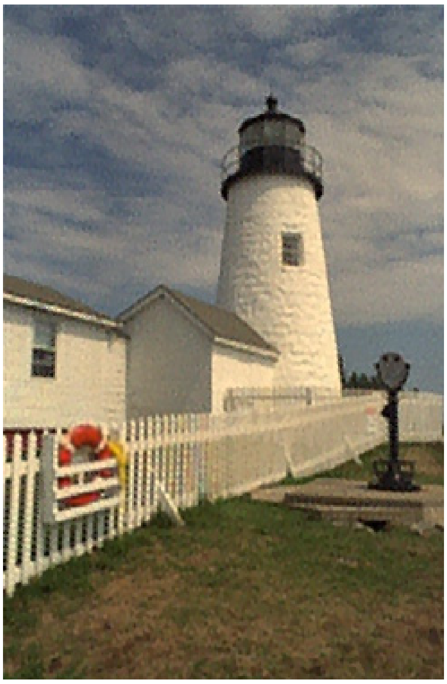
a:Reconst. from SR [4] on raw images

b: Reconst. with inter-color and lumin. regul.

c: Reconst. with chromin. and inter-color regul.

d: Reconst. from chromin. and lumin. regul.

Figure 3.9: The result of super-resolving each color band (raw data before demosaicing) separately considering only bilateral regularization [4], is shown in (a). Multi-frame demosaicing of this set of low-resolution frames with the help of only inter-color dependencies-luminance, inter-color dependencies-chrominance, and luminance-chrominance regularization terms is shown in (b), (c), and (d), respectively.



a: Reconst. with all terms.

Figure 3.10: The result of applying the proposed method (using all regularization terms) to this data set is shown in (a).

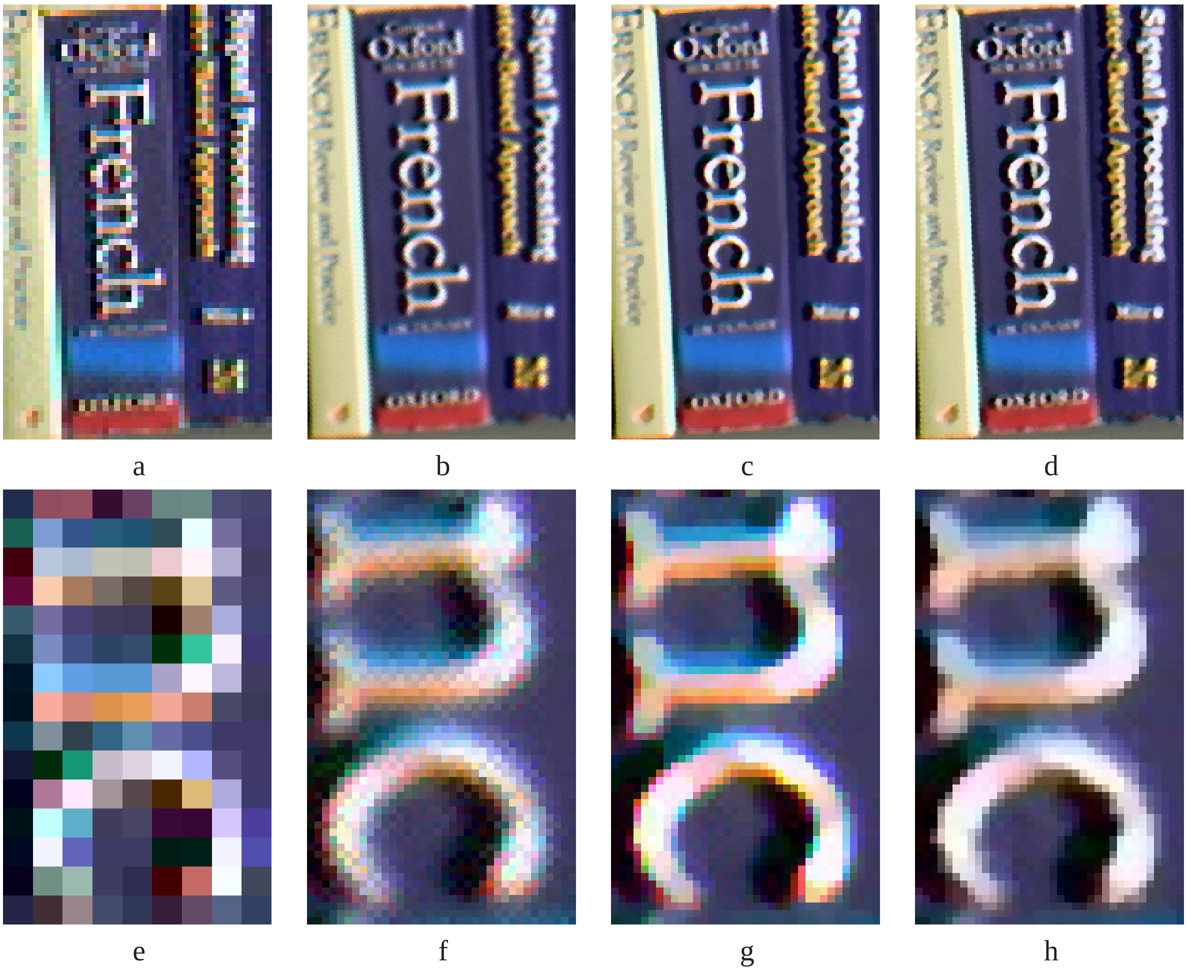
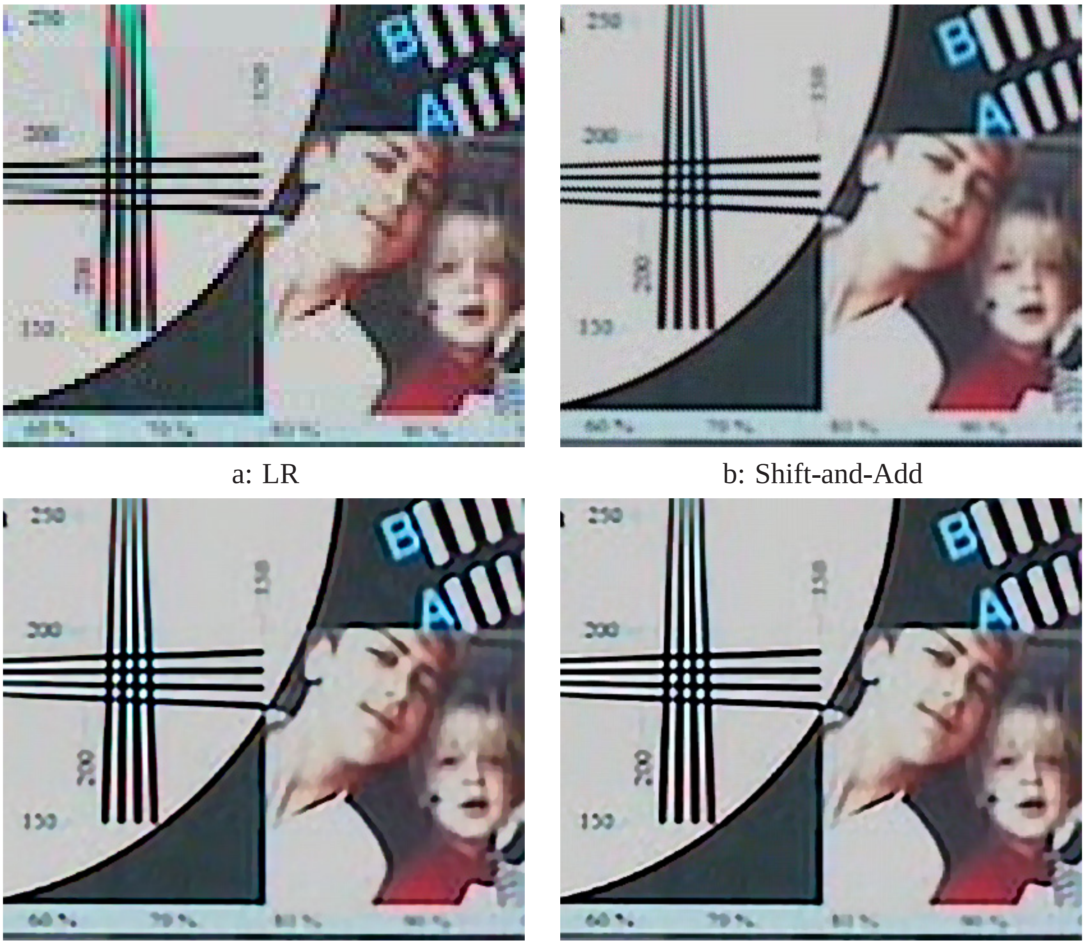


Figure 3.11: Multi-frame color super-resolution implemented on a real data (Bookcase) sequence. (a) shows one of the input low-resolution images of size $[75 \times 45 \times 3]$ and (b) is the Shift-and-Add result of size $[300 \times 180 \times 3]$, increasing resolution by a factor of 4 in each direction. (c) is the result of the individual implementation of the super-resolution [4] on each color band. (d) is implementation of (3.11) which has increased the spatial resolution, removed the compression artifacts, and also reduced the color artifacts. Figures (e) (of size $[15 \times 9 \times 3]$), (f) (of size $[60 \times 36 \times 3]$), (g), and (h) are the zoomed images of the Figures (a), (b), (c), and (d) respectively.



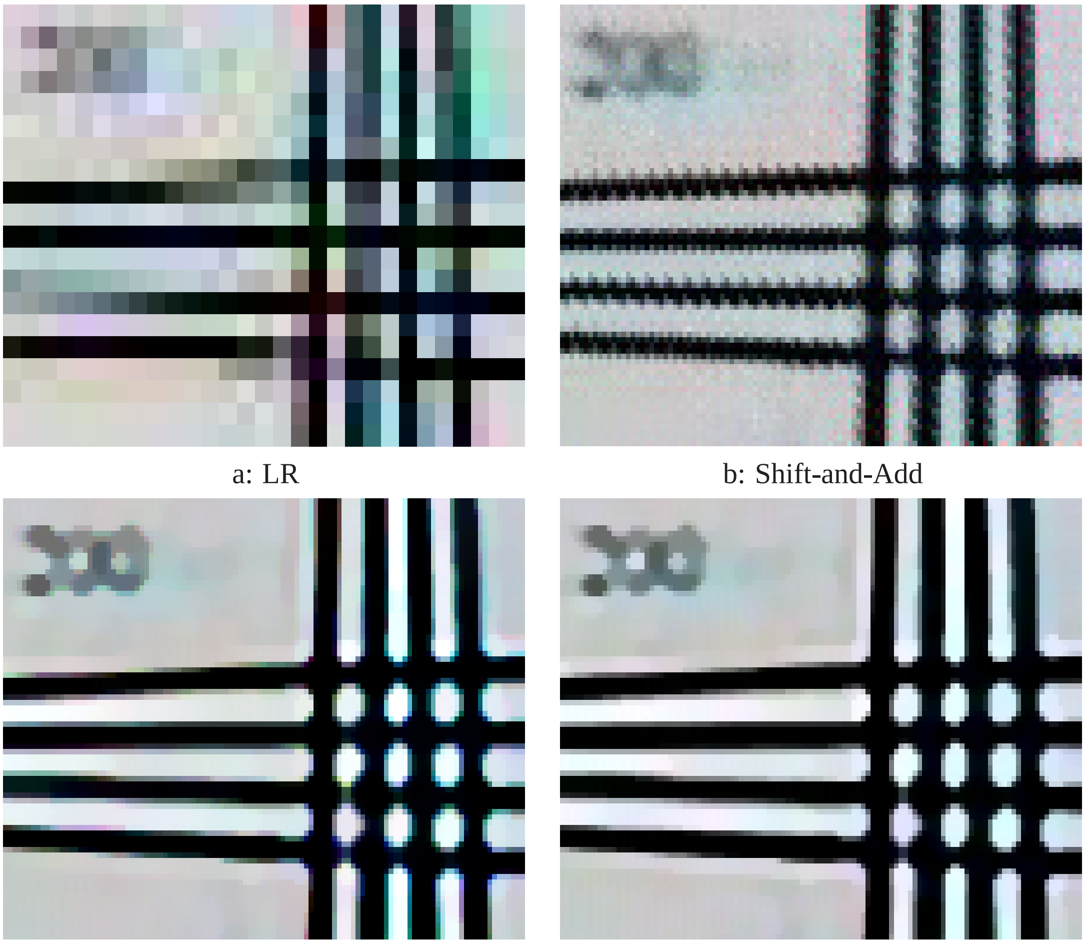
a: LR

b: Shift-and-Add

c: SR [4] on LR frames

d: Proposed method

Figure 3.12: Multi-frame color super-resolution implemented on a real data sequence. (a) shows one of the input low-resolution images (of size $[85 \times 102 \times 3]$) and (b) is the Shift-and-Add result (of size $[340 \times 408 \times 3]$) increasing resolution by a factor of 4 in each direction. (c) is the result of the individual implementation of the super-resolution [4] on each color band. (d) is implementation of (3.11) which has increased the spatial resolution, removed the compression artifacts, and also reduced the color artifacts. These images are zoomed in Figure 3.13.



a: LR

b: Shift-and-Add

c: SR [4] on LR frames

d: Proposed method

Figure 3.13: Multi-frame color super-resolution implemented on a real data sequence. A selected section of Figure 3.12(a), 3.12(b), 3.12(c), and 3.12(d) are zoomed in Figure 3.13(a) (of size $[20 \times 27 \times 3]$), 3.13(b) (of size $[80 \times 108 \times 3]$), 3.13(c), and 3.13(d), respectively. In (d) almost all color artifacts that are present on the edge areas of (a), (b), and (c) are effectively removed.

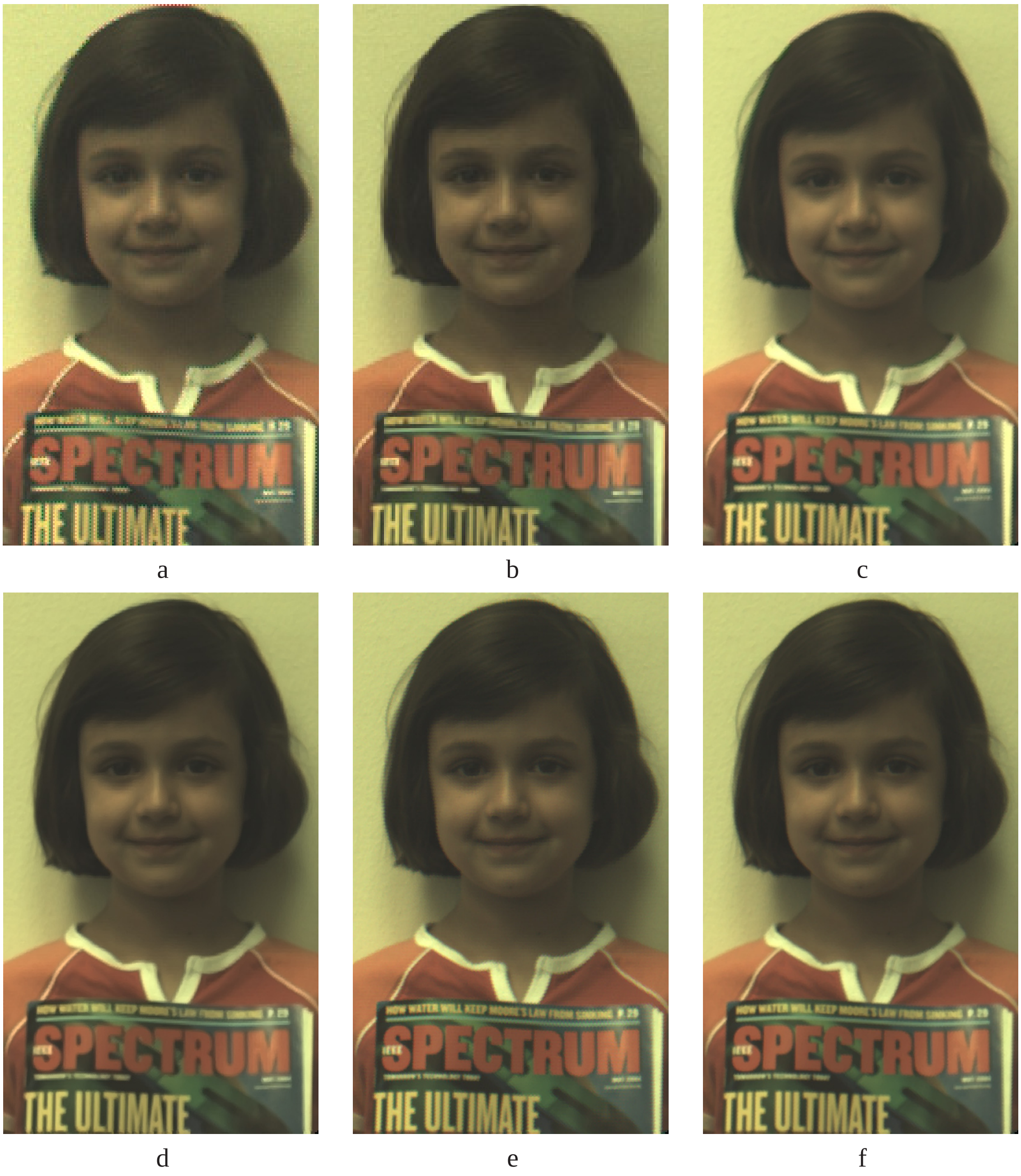


Figure 3.14: Multi-frame color super-resolution implemented on a real data sequence. (a) shows one of the input low-resolution images (of size $[290 \times 171 \times 3]$) demosaiced by [3] and (b) is one of the input low-resolution images demosaiced by the more sophisticated [2]. (c) is the result of applying the proposed color-super-resolution method on 31 low-resolution images each demosaiced by [3] method (high-resolution image of size $[870 \times 513 \times 3]$). (d) is the result of applying the proposed color-super-resolution method on 31 low-resolution images each demosaiced by [2] method. The result of applying our method on the original mosaiced raw low-resolution images (without using the inter color dependence term) is shown in (e). (f) is the result of applying our method on the original mosaiced raw low-resolution images.

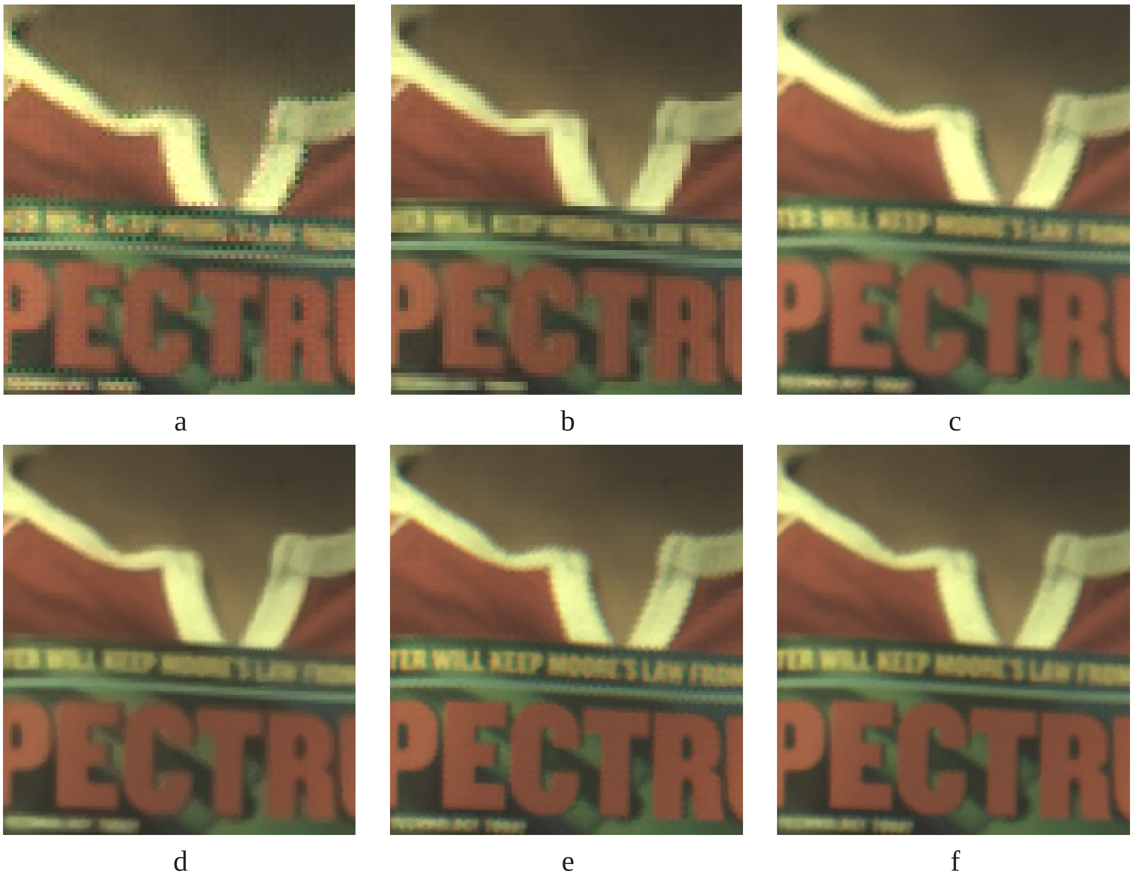


Figure 3.15: Multi-frame color super-resolution implemented on a real data sequence (zoomed). (a) shows one of the input low-resolution images (of size $[87 \times 82 \times 3]$) demosaiced by [3] and (b) is one of the input low-resolution images demosaiced by the more sophisticated [2]. (c) is the result of applying the proposed color-super-resolution method on 31 low-resolution images each demosaiced by [3] method (high-resolution image of size $[261 \times 246 \times 3]$). (d) is the result of applying the proposed color-super-resolution method on 31 low-resolution images each demosaiced by [2] method. The result of applying our method on the original mosaiced raw low-resolution images (without using the inter color dependence term) is shown in (e). (f) is the result of applying our method on the original mosaiced raw low-resolution images.

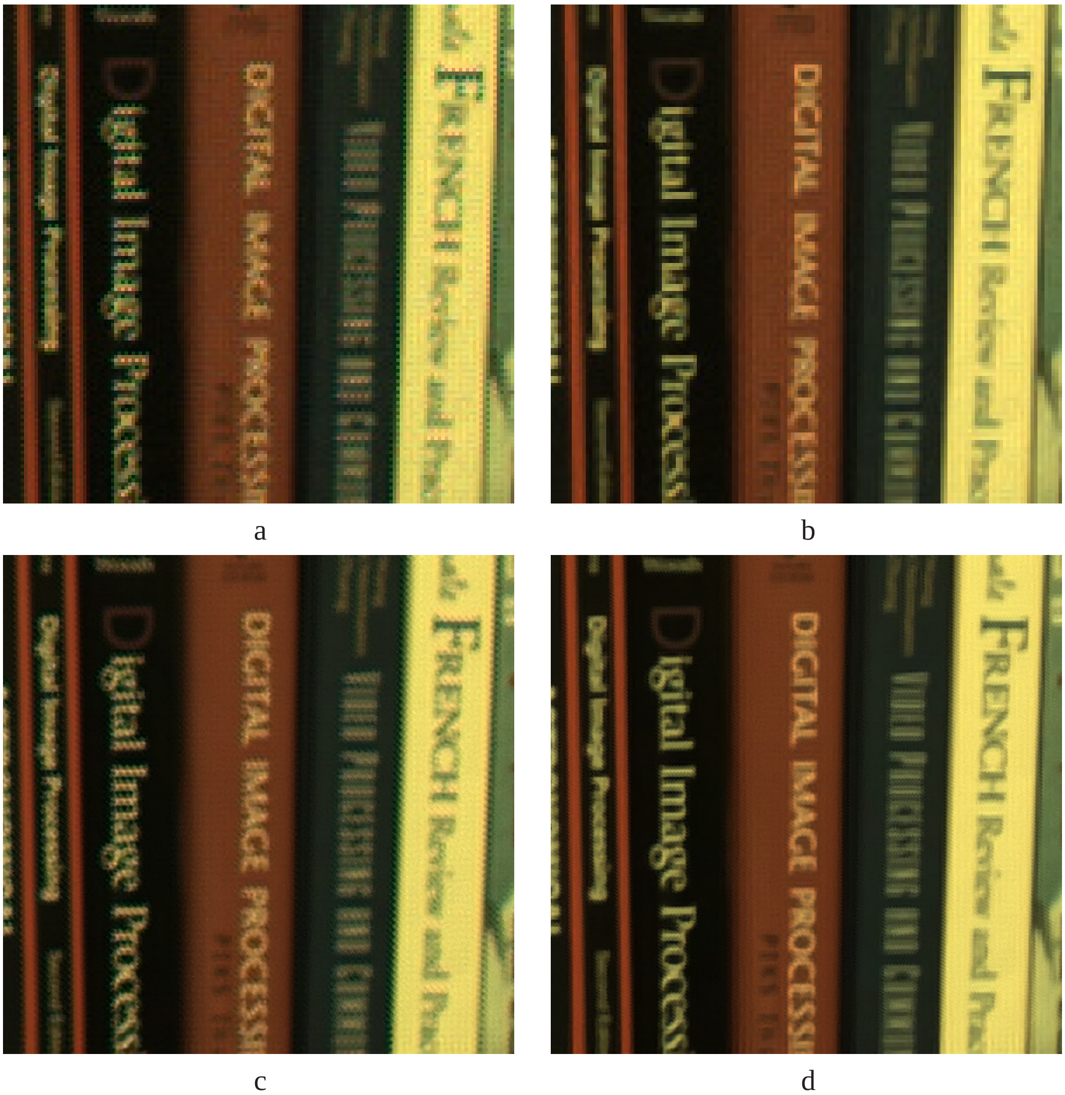
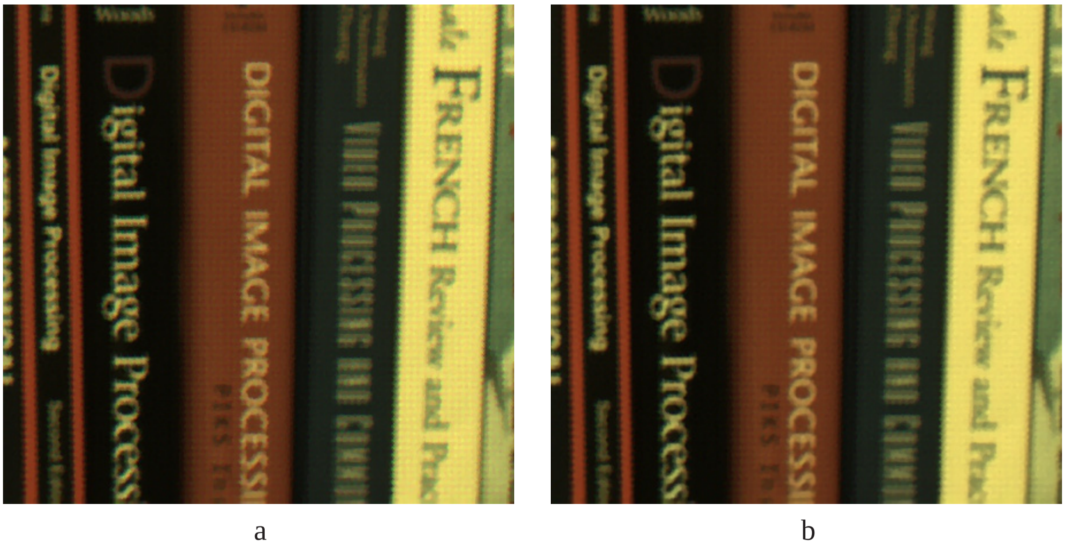


Figure 3.16: Multi-frame color super-resolution implemented on a real data sequence. (a) shows one of the input low-resolution images (of size $[141 \times 147 \times 3]$) demosaiced by [3] and (b) is one of the input low-resolution images demosaiced by the more sophisticated [2]. (c) is the result of applying the proposed color-super-resolution method on 31 low-resolution images each demosaiced by [3] method (high-resolution image of size $[423 \times 441 \times 3]$). (d) is the result of applying the proposed color-super-resolution method on 31 low-resolution images each demosaiced by [2] method.



a

b

Figure 3.17: Multi-frame color super-resolution implemented on a real data sequence. The result of applying our method on the original mosaiced raw low-resolution images (without using the inter color dependence term) is shown in (a) (high-resolution image of size $[423 \times 441 \times 3]$). (b) is the result of applying our method on the original mosaiced raw low-resolution images.

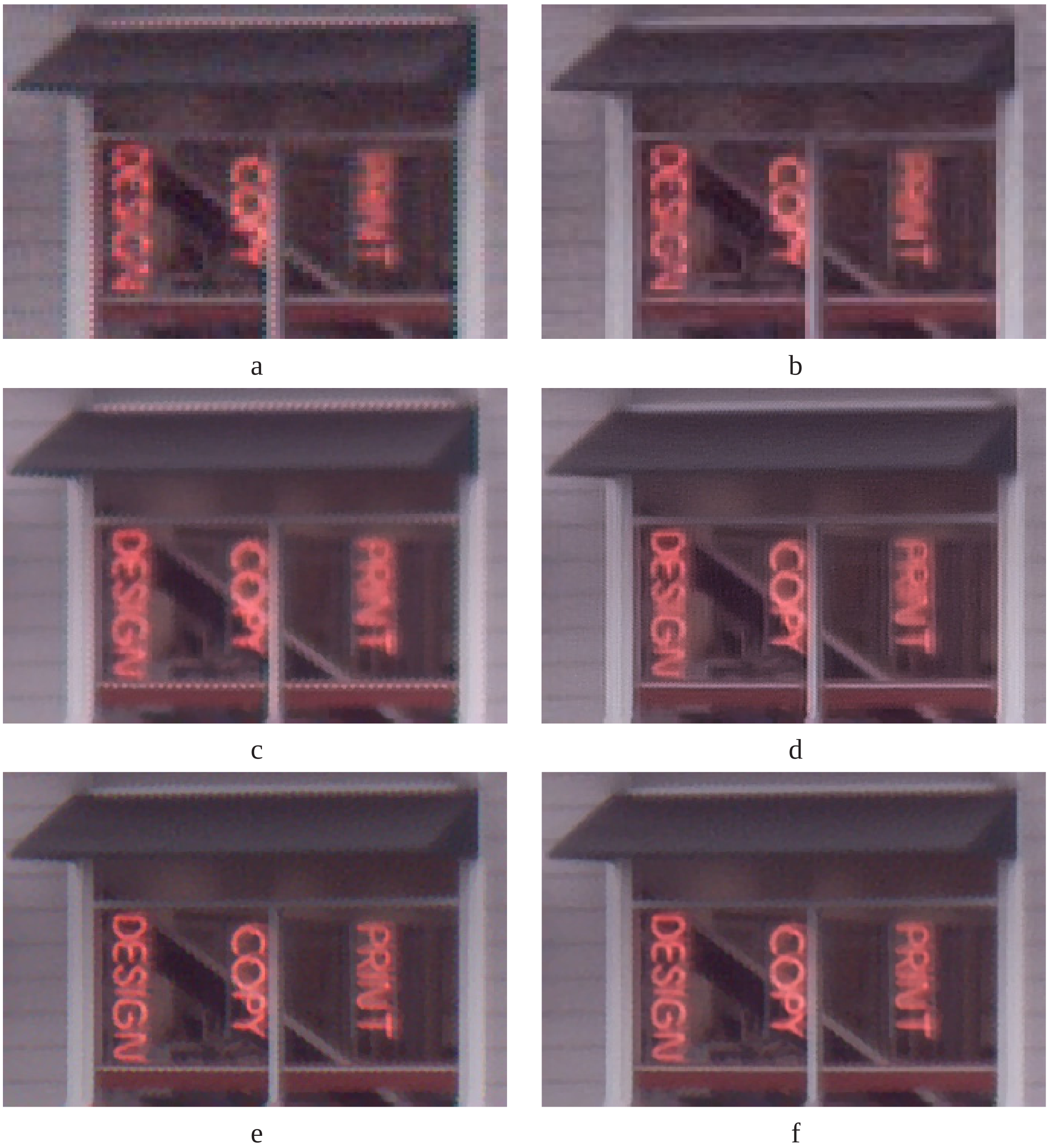


Figure 3.18: Multi-frame color super-resolution implemented on a real data sequence. (a) shows one of the input low-resolution images (of size $[81 \times 111 \times 3]$) demosaiced by [3] and (b) is one of the input low-resolution images demosaiced by the more sophisticated [2]. (c) is the result of applying the proposed color-super-resolution method on 30 low-resolution images each demosaiced by [3] method (high-resolution image of size $[243 \times 333 \times 3]$). (d) is the result of applying the proposed color-super-resolution method on 30 low-resolution images each demosaiced by [2] method. The result of applying our method on the original mosaiced raw low-resolution images (without using the inter color dependence term) is shown in (e). (f) is the result of applying our method on the original mosaiced raw low-resolution images.

Chapter 4

Dynamic Super-Resolution

4.1 Introduction

In the previous chapters, our approach to the super resolution problem was to fuse a set of low-resolution images and reconstruct a single high-resolution image. We refer to this as the *static* SR method, since it does not exploit the temporal evolution of the process. In this chapter, we consider SR applied on an image sequence, producing a sequence of high-resolution images; a process known as *dynamic* SR. The natural approach, as most existing works so far suggest, is to apply the static SR on a set of images with the t-th frame as a reference, produce the SR output, and repeat this process all over again per each temporal point. However, the memory and computational requirements for the static process are so taxing as to preclude its direct application to the dynamic case.

In contrast, in this chapter we adopt a dynamic point of view, as introduced in [73, 74], in developing the new SR solution. We take advantage of the fact that if the SR problem is solved for time $t-1$, our task for time t could use the solution at the previous time instant as a stepping stone towards a faster and more reliable SR solution. This is the essence of how dynamic SR is to gain its speed and better results, as compared to a sequence of detached static SR solutions.

The chapter presented here builds on the core ideas as appeared in [73, 74], but deviates from them in several important ways, to propose a new and better reconstruction algorithm:

- **Speed**: Whereas the methods in [73, 74] rely on the information-pair to approximate the Kalman Filter[1] (KF), this work uses the more classic mean-covariance approach. We show that for the case of translational motion and common space-invariant blur, the proposed method is computationally less complex than the dynamic SR methods proposed previously. Also, in line with Chapter 2, we show that this problem can be decomposed into two disjoint pieces, without sacrificing optimality.

- **Treatment of Mosaiced Images**: As introduced in the last chapter, two common resolution-enhancement problems in digital video/photography that are typically addressed separately--namely, SR and demosaicing can be treated jointly. In this chapter, we propose a method of dealing with these two problems jointly, and *dynamically*. Note that in the previous chapter as in [66, 75] we addressed the static multi-frame demosaicing problem, and so the work presented here stands as an extension of it to the dynamic case.

- **Treatment of Color**: Our goal in this chapter is to develop a dynamic SR algorithm for both monochromatic and color input and output sequences. We seek improvements in both visual quality (resolution enhancement and color artifact reduction) and computational/memory efficiency.

- **Causality**: The work presented in [73, 74] considered a causal mode of operation, where the output image at time t_0 fuses the information from times $t \leq t_0$. This is the appropriate mode of operation when on-line processing is considered. Here, we also study a non-causal processing mode, where every high-resolution reconstructed image is derived as an optimal estimate incorporating information from *all* the frames in the sequence. This is an appropriate mode of operation for off-line processing of movies, stored on disk. We use the smoothing KF formulation to obtain an efficient algorithm for this case.

[1] Kalman filtering is the best linear unbiased estimation (BLUE) technique for recovering sequential signals contaminated with additive noise. Kalman estimators have desirable computational properties for dynamic systems, since they recursively update their estimated signal based on the new arrived data and the previous estimated signal and its corresponding covariance [41].

This chapter is organized as follows: In Section 4.2 we discuss a fast dynamic image fusion method for the translational motion model, assuming regular monochromatic images, considering both causal and non-causal modes. This method is then extended in Section 4.3 to consider an enhancement algorithm of monochromatic deblurring and interpolation. We address multi-frame demosaicing and color-SR deblurring problems in Section 4.4. Simulations on both real and synthetic data sequences are presented in Section 4.5; and Section 4.6 concludes this chapter.

4.2 Dynamic Data Fusion

4.2.1 Recursive Model

In this chapter, we use the general linear dynamic forward model for the SR problem as in [73, 74]. A dynamic scene with intensity distribution $\underline{X}(t)$ is seen to be warped at the camera lens because of the relative motion between the scene and camera, and blurred by camera lens and sensor integration. Then, it is discretized at the CCD, resulting in a digitized noisy frame $\underline{Y}(t)$. Discretization in many commercial digital cameras is a combination of color filtering and down-sampling processes. However, in this section we shall restrict our treatment to simple monochrome imaging. We represent this forward model by the following state-space equations [41]:

$$\underline{X}(t) = F(t)\underline{X}(t-1) + \underline{U}(t), \tag{4.1}$$

and

$$\underline{Y}(t) = D(t)H(t)\underline{X}(t) + \underline{V}(t). \tag{4.2}$$

Equation (4.1) describes how the ideal super-resolved images relate to each other through time. Similar to Chapter 2, we use the underscore notation such as $\underline{X}$ to indicate a vector derived from the corresponding image of size $[rQ_1 \times rQ_2]$ pixels, scanned in lexicographic order. The current image $\underline{X}(t)$ is of size $[r^2Q_1Q_2 \times 1]$, where r is the resolution enhancement factor, and $[Q_1 \times Q_2]$ is the size of an input low-resolution image. Equation (4.1) states that up to some innovation content $\underline{U}(t)$, the current high-resolution image is a geometrically warped version of the previous image, $\underline{X}(t-1)$. The $[r^2Q_1Q_2 \times r^2Q_1Q_2]$ matrix $F(t)$ represents this warp operator. The so-called system noise $\underline{U}(t)$, of size $[r^2Q_1Q_2 \times 1]$, is assumed to be additive zero mean Gaussian with $C_u(t)$ as its covariance matrix of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$. Note that the closer the overlapping regions of $\underline{X}(t)$ and the motion compensated $\underline{X}(t-1)$ are, the smaller $C_u(t)$ becomes. Therefore, $C_u(t)$ in a sense reflects the accuracy of the motion estimation process (as $\underline{U}(t)$ also captures unmodeled motions), and for overlapped regions it is directly related to the motion estimation covariance matrix.

As to Equation (4.2), it describes how the measured image $\underline{Y}(t)$ of size $[Q_1Q_2 \times 1]$ is related to the ideal one, $\underline{X}(t)$. The camera's point spread function (PSF) is modeled by the $[r^2Q_1Q_2 \times r^2Q_1Q_2]$ blur matrix $H(t)$, while the $[Q_1Q_2 \times r^2Q_1Q_2]$ matrix $D(t)$ represents the down-sampling operation at the CCD (down-sampling by the factor r in each axis). In mosaiced cameras, this matrix also represents the effects of the color filter array, which further down-samples the color images – this will be described and handled in Section 4.4. The noise vector $\underline{V}(t)$ of size $[Q_1Q_2 \times 1]$ is assumed to be additive, zero mean, white Gaussian noise. Thus, its $[Q_1Q_2 \times Q_1Q_2]$ covariance matrix is $C_v(t) = \sigma_v^2 I$. We further assume that $\underline{U}(t)$ and $\underline{V}(t)$ are independent of each other.

The equations given above describe a system in its *state-space* form, where the state is the desired ideal image. Thus, a KF formulation can be employed to recursively compute the optimal estimates $(\underline{X}(t), t \in \{1, ..., N\})$ from the measurements $(\underline{Y}(t), t \in \{1, ..., N\})$, assuming that $D(t), H(t),\ F(t), \sigma_v,$ and $C_u(t)$ are all known [41, 73, 74]. This estimate could be done causally, as an on-line processing of an incoming sequence, or non-causally, assuming that the entire image sequence is stored on disk and processed off-line. We consider both options in this chapter.

As to the assumption about the knowledge of various components of our model, while each of the operators $D(t), H(t),$ and $F(t)$ may vary in time, for most situations the down-sampling (and later color filtering), and camera blurring operations remain constant over time, assuming that the images are obtained from the same camera. In this chapter, we further assume that the camera PSF is space-invariant, and the motion is composed of pure translations, accounting for either vibrations of a gazing camera, or a panning motion of a camera viewing a far away scene. Thus, both H and $F(t)$ are block-circulant matrices[2], and as such, they commute. We assume that H is known, being dependent on the camera used, and $F(t)$ is built from motion estimation applied on the raw sequence $\underline{Y}(t)$. The down-sampling operator D is completely dictated by the choice of the resolution enhancement factor (r). As to σ_v, and $C_u(t)$, those will be handled shortly.

We limit our model to the case of translational motion for several reasons. First, as we describe later, such a motion model allows for an extremely fast and memory efficient dynamic SR algorithm. Second, while simple, the model fairly well approximates the motion contained in many image sequences, where the scene is stationary and only the camera moves in approximately linear fashion. Third, for sufficiently high frame rates most motion models can be (at least locally) approximated by the translational model. Finally, we believe that an in-depth study of this simple case yields much insight into the more general cases of motion in dynamic SR.

By substituting $\underline{Z}(t) = H\underline{X}(t)$, we obtain from (4.1) and (4.2) an alternative model, where the state vector is $\underline{Z}(t)$,

$$\underline{Z}(t) = F(t)\underline{Z}(t-1) + \underline{W}(t), \tag{4.3}$$

and

$$\underline{Y}(t) = D\underline{Z}(t) + \underline{V}(t). \tag{4.4}$$

Note that the first of the two equations is obtained by left multiplication of both sides of (4.1) by H and using the fact that it commutes with $F(t)$. Thus, the vector $\underline{W}(t)$ is a colored version of $\underline{U}(t)$, leading to $C_w(t) = HC_u(t)H^T$ as the covariance matrix.

[2] True for cyclic boundary conditions, that will be assumed throughout this work.

With this alternative definition of the state of the dynamic system, the solution of the inverse problem at hand decomposes, without loss of optimality, into the much simpler subtasks of fusing the available images to compute the estimated blurry image$\hat{\underline{Z}}(t)$, followed by a deblurring/interpolation step, estimating$\hat{\underline{X}}(t)$ from $\hat{\underline{Z}}(t)$. In this section, we treat the three color bands separately. For instance, only the red band values in the input frames,$\underline{Y}(t)$, contribute to the reconstruction of the red band values in$\hat{\underline{Z}}(t)$. The correlation of the different color bands are discussed and exploited in Section 4.4.

We next study the application of KF to estimate $\underline{Z}(t)$. In general, the application of KF requires the update of the state-vector's covariance matrix per each temporal point, and this update requires an inversion of the state-vector's covariance matrix. For a super-resolved image with $r^2Q_1Q_2$ pixels, this matrix is of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$, implying a prohibitive amount of computations and memory.

Fast and memory efficient alternative ways are to be found, and such methods were first proposed in the context of the dynamic SR in [73, 74]. Here we show that significant further speedups are achieved for the case of translational motion and common space-invariant blur.

4.2.2 Forward Data Fusion Method

The following defines the forward Kalman propagation and update equations [41], that account for a causal (on-line) process. We assume that at time $t-1$ we already have the mean-covariance pair, $(\hat{\underline{Z}}(t-1), \hat{\Pi}(t-1))$, and those should be updated to account for the information obtained at time t. We start with the covariance matrix update based on Equation (4.3),

$$\tilde{\Pi}(t) = F(t)\hat{\Pi}(t-1)F^T(t) + C_{w(t)}, \tag{4.5}$$

where $\tilde{\Pi}(t)$ is the propagated covariance matrix (initial estimate of the covariance matrix at time t). The KF gain matrix is given by

$$K(t) = \tilde{\Pi}(t)D^T[C_v(t) + D\tilde{\Pi}(t)D^T]^{-1}. \tag{4.6}$$

This matrix is rectangular of size $[r^2Q_1Q_2 \times Q_1Q_2]$. Based on $K(t)$, the updated state vector mean is computed by

$$\underline{\hat{Z}}(t) = F(t)\underline{\hat{Z}}(t-1) + K(t)[\underline{Y}(t) - DF(t)\underline{\hat{Z}}(t-1)]. \tag{4.7}$$

The final stage requires the update of the covariance matrix, based on Equation (4.4),

$$\hat{\Pi}(t) = \text{Cov}\left(\underline{\hat{Z}}(t)\right) = [\mathbf{I} - K(t)D]\tilde{\Pi}(t). \tag{4.8}$$

More on the meaning of these equations and how they are derived can be found in [41, 76].

While in general the above equations require the propagation of intolerably large matrices in time, if we refer to $C_w(t)$ as a diagonal matrix, then $\tilde{\Pi}(t)$, and $\hat{\Pi}(t)$ are diagonal matrices of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$. It is relatively easy to verify this property: for an arbitrary diagonal matrix G_B (B stands for *big*), the matrix DG_BD^T is a diagonal matrix. Similarly, for an arbitrary diagonal matrix G_S (S stands for *small*), the matrix D^TG_SD is diagonal as well. Also, in [22] it is shown that for an arbitrary pure translation matrix F and an arbitrary diagonal matrix G_B, the matrix FG_BF^T is diagonal. Therefore, if the matrix $\tilde{\Pi}(0)$ is initialized as a diagonal matrix, then $\tilde{\Pi}(t)$, and $\hat{\Pi}(t)$ are necessarily diagonal for all t, being the results of summation, multiplication, and inversions of diagonal matrices.

Diagonality of $C_w(t)$ is a key assumption in transferring the general KF into a simple and fast procedure, and as we shall see, the approximated version emerging is quite faithful. Following [73, 74], if we choose a matrix $\sigma_w^2\mathbf{I} \geq C_w(t)$, it implies that $\sigma_w^2\mathbf{I} - C_w(t)$ is a positive semi-definite matrix, and there is always a finite σ_w that satisfies this requirement. Replacing $C_w(t)$ with this majorizing diagonal matrix, the new state-space system in Equations (4.3) and (4.4) simply assumes a stronger innovation process. The effect on the KF is to rely less on the temporal relation in (4.3) and more on the measurements in (4.4). In fact, at the extreme case, if $\sigma_w \rightarrow \infty$, the KF uses only the measurements, leading to an intra-frame maximum-likelihood estimator. Thus, more generally, such a change causes a loss in the accuracy of the KF because it relies less on the internal dynamics of the system, but this comes with a welcomed simplification of the recursive estimator. It must be clear that such change in $C_w(t)$

$$\underset{G_S}{\begin{array}{|c|c|c|}\hline a & 0 & 0 \\ \hline 0 & b & 0 \\ \hline 0 & 0 & c \\ \hline \end{array}} \quad \begin{array}{c} D^T G_S D \\ \Longrightarrow \\ \Longleftarrow \\ D G_B D^T \end{array} \quad \underset{G_B}{\begin{array}{|c|c|c|c|c|c|}\hline a & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & b & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & c & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}}$$

Figure 4.1: The diagonal matrix G_B on the right is the result of applying the up-sampling operation ($D^T G_S D$) on an arbitrary diagonal matrix G_S on the left. The matrix G_S can be retrieved by applying the down-sampling operation ($DG_B D^T$). The up-sampling/down-sampling factor for this example is two.

has no impact on the convergence properties of the dynamic estimator we apply, and it does not introduce a bias in the estimate. Note that all the above is true also for a diagonal non-toeplitz alternative, where the main diagonal entries are varying is space.

Once we chose $C_w(t)$ to be diagonal, Equations (4.5), (4.6), (4.7), and (4.8) are simplified, and their use is better understood on a pixel-by-pixel basis. Before we turn to describe such a KF for the forward case, we introduce some notations to simplify the explanation of the process.

The warp matrix $F(t)$ and its transpose can be exactly interpreted as image shift operators [4, 22]. We use hereafter the superscript "f", to simplify the notation of forward shifting of vectors and diagonal matrices, and thus, $\underline{Z}^f(t) = F(t)\underline{Z}(t-1)$ and $\hat{\Pi}^f(t) = F(t)\hat{\Pi}(t-1)F^T(t)$.

Also, the matrix D and its transpose can be exactly interpreted as down-sampling and up-sampling operators. Application of $D\underline{Z}(t)$ and $D\hat{\Pi}(t)D^T$ results in down-sampling of the vector $\underline{Z}(t)$ and the diagonal matrix $\hat{\Pi}(t)$. Likewise, application of $D^T\underline{Y}(t)$ and $D^T C_v(t)D$ results in up-sampling of the vector $\underline{Y}(t)$ and the diagonal matrix $C_v(t)$ with zero filling. Figure 4.1 illustrates the effect of matrix up-sampling and down-sampling operations, and this also sheds some light on the previous discussion on the diagonality assumption on$\tilde{\Pi}(t)$ and $\hat{\Pi}(t)$.

Finally, we will use the notation $[G]_q$ to refer to the (q, q) entry of the diagonal matrix G, and $[\underline{G}]_q$ to refer to the $(q, 1)$ entry in the vector $\underline{G}$. This way we will be able to handle both the low-resolution and the high-resolution grids in the same equations. Let us now return to the KF equations and show how they are implemented in practice on a pixel-by-pixel basis. First, referring to the propagated covariance matrix, we start by observing that in Equation (4.6), the term $C_v(t) + D\tilde{\Pi}D^T$ is a diagonal matrix of size $[Q_1Q_2 \times Q_1Q_2]$, with the (q, q)-th entry being

$$[C_v(t)]_q + [\hat{\Pi}^f(t)]_{qr^2} + [C_w(t)]_{qr^2},$$

with q in the range $[1, Q_1Q_2]$. The "jumps" in r^2 in the indices of $\hat{\Pi}^f(t)$ and $C_w(t)$ are caused by the decimation D. Applying an inversion replaces the above by its reciprocal. Using interpolation $D^T(C_v(t) + D\tilde{\Pi}D^T)^{-1}D$ gives a diagonal matrix of size $[r^2Q_1Q_2 \times r^2Q_1Q_2]$, with the q-th entry being

$$\frac{1}{[C_v(t)]_{\frac{q}{r^2}} + [\hat{\Pi}^f(t)]_q + [C_w(t)]_q},$$

this time referring to the indices $q = r^2,\ 2r^2,\ \ldots\ , Q_1Q_2r^2$. For all other $(r^2 - 1)Q_1Q_2$ indices, the entries are simply zeros, filled by the interpolation. Merging this with Equations (4.6) and (4.8), we obtain

$$[\hat{\Pi}(t)]_q = \begin{cases} \frac{[C_v(t)]_{\frac{q}{r^2}}\left([\hat{\Pi}^f(t)]_q + [C_w(t)]_q\right)}{[C_v(t)]_{\frac{q}{r^2}} + [\hat{\Pi}^f(t)]_q + [C_w(t)]_q} & \text{for } q = r^2,\ 2r^2,\ \ldots\ , Q_1Q_2r^2 \\ [\hat{\Pi}^f(t)]_q + [C_w(t)]_q & \text{otherwise}. \end{cases} \tag{4.9}$$

Note that the incorporation of each newly measured low-resolution image only updates values of Q_1Q_2 entries in the diagonal of $\hat{\Pi}(t)$, located at the $[r^2, 2r^2, .., r^2Q_1Q_2]$ positions. The remaining $(r^2 - 1)Q_1Q_2$ diagonal entries are simply propagated from the previous temporal point, based on Equation (4.5) only. As we shall see, the same effect holds true for the update of $\hat{\underline{Z}}(t)$, where $(r^2 - 1)Q_1Q_2$ entries are propagated from the previous temporal point without an update.

Turning to the update of the mean vector, $\hat{\underline{Z}}(t)$, using the same reasoning applied on

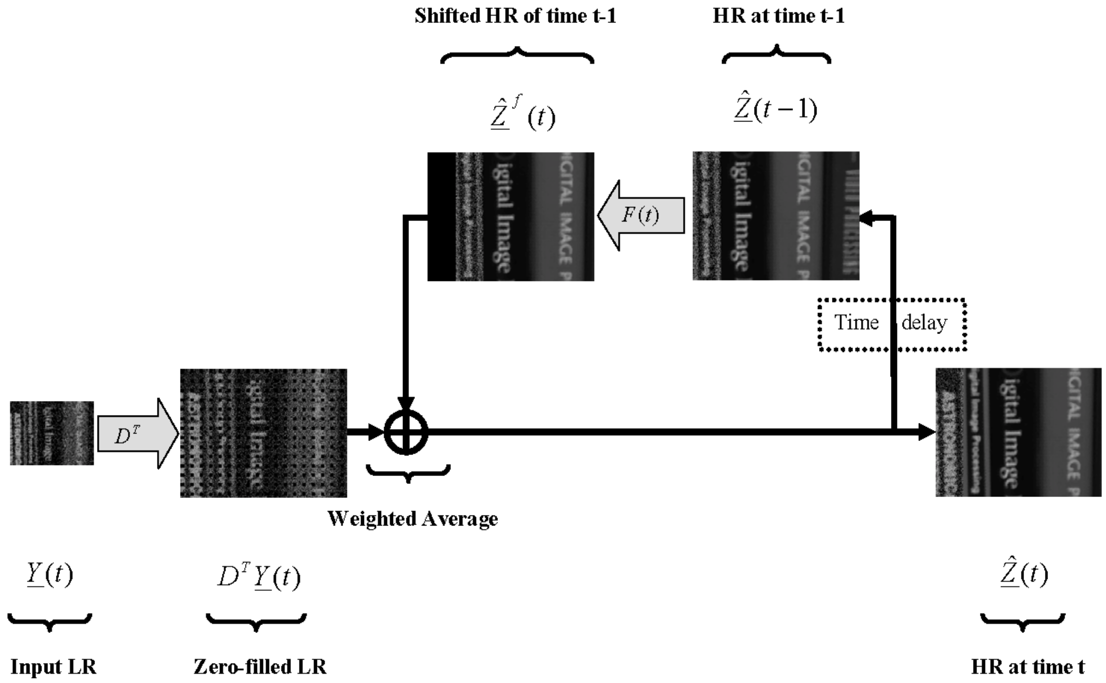


Figure 4.2: Block diagram representation of (4.10), where $\underline{\hat{Z}}(t)$, the new input high-resolution output frame is the weighted average of $\underline{Y}(t)$, the current input low-resolution frame and $\underline{\hat{Z}}^f(t)$, the previous estimate of the high-resolution image after motion compensation.

Equations (4.6) and (4.7), we obtain the relation

$$[\underline{\hat{Z}}(t)]_q = \begin{cases} \frac{[C_v(t)]_{\frac{q}{r^2}}[\underline{\hat{Z}}^f(t)]_q + \left([\hat{\Pi}^f(t)]_q + [C_w(t)]_q\right)[\underline{Y}(t)]_{\frac{q}{r^2}}}{[C_v(t)]_{\frac{q}{r^2}} + [\hat{\Pi}^f(t)]_q + [C_w(t)]_q} & \text{for } q = r^2,\ 2r^2,\ \ldots\ , Q_1Q_2r^2 \\ [\underline{\hat{Z}}^f(t)]_q & \text{otherwise}. \end{cases} \tag{4.10}$$

Figure 4.2 describes the above equation's upper part as a block diagram. Notice that two images are merged here – an interpolated version of$\underline{Y}(t)$, and$\underline{\hat{Z}}^f(t)$. The merging is done as a weighted average between the two, as the figure suggests.

The overall procedure using these update equations is outlined below in Algorithm 1. Since the update operations are simply based on shifting the previous estimates$\underline{\hat{Z}}(t-1)$ and $\hat{\Pi}(t-1)$ and updating the proper pixels using (4.9) and (4.10), we refer hereafter to this algorithm as the dynamic Shift-and-Add process. Similarly, we call$\underline{\hat{Z}}(t)$ the dynamic Shift-and-Add image. Several comments are in order, regarding the above procedure:

1. Initialization: For long enough sequences, the initialization choice has a vanishing effect on the outcome. Choosing $\hat{\Pi}(0) = \epsilon^2\mathbf{I}$ guarantees that $\hat{\Pi}(t)$ is strictly positive definite at all times, where ϵ is an arbitrary large number ($\epsilon \gg \sigma_v^2$). Better initialization can be proposed, based on interpolation of the image $\underline{Y}(t)$. The same applies to regions coming from occlusion – those can be initialized by the current image.

2. Arrays propagated in time: The algorithm propagates two images in time--namely, the image estimate $\hat{\underline{Z}}(t)$, and the main diagonal of its covariance matrix $\hat{\Pi}(t)$. This last quantity represents the weights assigned per pixel for the temporal fusion process, where the weights are derived from the accumulated measurements for the pixel in question.

```
• Task: Given $\{\underline{Y}(t)\}_{t\geq 1}$, estimate $\{\underline{Z}(t)\}_{t\geq 1}$ causally.

• Initialization: Set $t = 0$, choose $\hat{\underline{Z}}(t) = \underline{0}$ and $\hat{\Pi}(t) = \epsilon^2\mathbf{I}$.

• Update Process: Set $t \to t + 1$, obtain $\underline{Y}(t)$ and apply

    1. Motion Compensation: Compute $\hat{\underline{Z}}^f(t) = F(t)\hat{\underline{Z}}(t-1)$ and
       $\hat{\Pi}^f(t) = F(t)\hat{\Pi}(t-1)F^T(t)$.

    2. Update of the Covariance: Use Equation (4.9) to
       compute the update $\hat{\Pi}(t)$.

    3. Update of the Mean: Use Equation (4.10) to compute
       the update $\hat{\underline{Z}}(t)$.

• Repeat: Update process.
```

Algorithm 1: Forward dynamic Shift-and-Add algorithm.

At this point, we have an efficient recursive estimation algorithm producing estimates of the blurry high-resolution image sequence $\hat{\underline{Z}}(t)$. From these frames, the sequence $\hat{\underline{X}}(t)$ should be estimated. Note that some (if not all) frames will not have estimates for every pixel in

$\underline{\hat{Z}}(t)$, necessitating a further joint interpolation and deblurring step, which will be discussed in Sections 4.3 and 4.4. For the cases of multi-frame demosaicing and color SR, the above process is to be applied separately on the R, G, and B layers, producing the arrays we will start from in the next sections.

While the recursive procedure outlined above will produce the optimal (minimum mean-squared) estimate of the state (blurry image $\underline{\hat{Z}}(t)$) in a causal fashion, we can also consider the best estimate of the same given "all" the frames. This optimal estimate is obtained by a two-way recursive filtering operation known as "smoothing", which we discuss next.

4.2.3 Smoothing Method

The fast and memory-efficient data fusion method described above is suitable for causal, real-time processing, as it estimates the high-resolution frames from the previously seen low-resolution frames. However, often times super-resolution is preformed off-line, and therefore a more accurate estimate of a high-resolution frame at a given time is possible by using both previous and future low-resolution frames. In this section, we study such an off-line dynamic SR method also known as smoothed dynamic SR [77].

The smoothed data fusion method is a two-pass (forward-backward) algorithm. In the first pass, the low-resolution frames pass through a forward data fusion algorithm similar to the method explained in Section 4.2.2, resulting in a set of high-resolution estimates $\{\underline{\hat{Z}}(t)\}_{t=1}^{N}$ and their corresponding diagonal covariance matrices $\{\hat{\Pi}(t)\}_{t=1}^{N}$. The second pass runs backward in time using those mean-covariance pairs, and improves these forward high-resolution estimates, resulting in the smoothed mean-covariance pairs $\{\underline{\hat{Z}}_s(t), \hat{\Pi}_s(t)\}_{t=1}^{N}$.

While it is possible to simply implement the second pass (backward estimation) similar to the forward KF algorithm, and obtain the smooth estimate by weighted averaging of the forward and backward estimates with respect to their covariance matrices, computationally more efficient methods are more desirable. In what follows we study such algorithm based on the fixed-interval smoothing method of Rauch-Tung-Striebel [78, 79].

The following equations define the high-resolution image and covariance updates in the second pass. Assuming that we have the entire (forward-filtered) sequence $\{\hat{\underline{Z}}(t), \hat{\Pi}(t)\}_{t=1}^{N}$, we desire to estimate the pairs $\{\hat{\underline{Z}}_s(t), \hat{\Pi}_s(t)\}_{t=1}^{N}$ that represent the mean and covariance per time t, based on all the information in the sequence. We assume a process that runs from $t = N - 1$ downwards, initialized with $\hat{\underline{Z}}_s(N) = \hat{\underline{Z}}(N)$ and $\hat{\Pi}_s(N) = \hat{\Pi}(N)$.

We start with the covariance propagation matrix. Notice its similarity to Equation (4.5):

$$\tilde{\Pi}(t+1) = F(t+1)\hat{\Pi}(t)F^T(t+1) + C_w(t+1). \tag{4.11}$$

This equation builds a prediction of the covariance matrix for time $t+1$, based on the first pass forward stage. Note that the outcome is diagonal as well.

The Kalman smoothed gain matrix is computed using the above prediction matrix, and the original forward covariance one, by

$$K_s(t) = \hat{\Pi}(t)F^T(t+1)[\tilde{\Pi}(t+1)]^{-1}. \tag{4.12}$$

This gain will be used both for the backward updates of the mean and the covariance,

$$\hat{\underline{Z}}_s(t) = \hat{\underline{Z}}(t) + K_s(t)[\hat{\underline{Z}}_s(t+1) - F(t+1)\hat{\underline{Z}}(t)], \tag{4.13}$$

where the term $\hat{\underline{Z}}_s(t+1) - F(t+1)\hat{\underline{Z}}(t)$ could be interpreted as a prediction error. The smoothed covariance matrix is updated by

$$\hat{\Pi}_s(t) = \text{Cov}\left(\hat{\underline{Z}}_s(t)\right) = \hat{\Pi}(t) + K_s(t)[\hat{\Pi}_s(t+1) - \tilde{\Pi}(t+1)]K_s^T(t). \tag{4.14}$$

Following the notations we have used before, we use the superscript "b" to represent backward shifting in time of vectors and matrices, so that $\hat{\underline{Z}}_s^b(t) = F^T(t+1)\hat{\underline{Z}}_s(t+1)$ and similarly $\hat{\Pi}_s^b(t) = F^T(t+1)\hat{\Pi}_s(t+1)F(t+1)$, and $C_w^b(t) = F^T(t+1)C_w(t+1)F(t+1)$. Then, using the same rationale practiced in the forward algorithm, the smoothed gain matrix for a pixel at spatial position q is

$$\frac{[\hat{\Pi}(t)]_q}{[\hat{\Pi}(t)]_q + [C_w^b(t)]_q}.$$

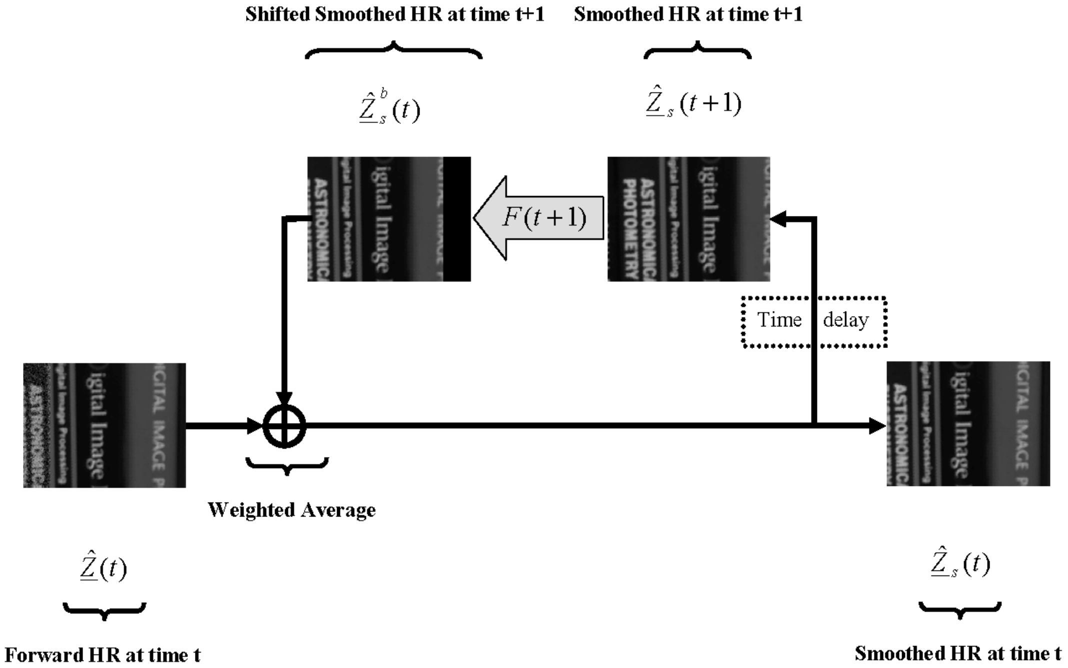


Figure 4.3: Block diagram representation of (4.16), where $\hat{\underline{Z}}_s(t)$, the new Rauch-Tung-Striebel smoothed high-resolution output frame is the weighted average of $\hat{\underline{Z}}(t)$, the forward Kalman high-resolution estimate at time t, and $\hat{\underline{Z}}^b_s(t)$, the previous smoothed estimate of the high-resolution image ($\hat{\underline{Z}}^b_s(t) = F^T(t+1)\hat{\underline{Z}}_s(t+1)$), after motion compensation.

Similar to what is shown in Section 4.2.2, we can simplify Equations (4.11), (4.12), (4.13), and (4.14) to the following pixel-wise update formulas

$$[\hat{\Pi}_s(t)]_q = [\hat{\Pi}(t)]_q + [\hat{\Pi}(t)]_q^2 \frac{[\hat{\Pi}^b_s(t)]_q - [\hat{\Pi}(t)]_q - [C^b_w(t)]_q}{[\hat{\Pi}(t)]_q + [C^b_w(t)]_q}, \tag{4.15}$$

and

$$[\hat{\underline{Z}}_s(t)]_q = \frac{[C^b_w(t)]_q[\hat{\underline{Z}}(t)]_q + [\hat{\Pi}(t)]_q[\hat{\underline{Z}}^b_s(t)]_q}{[\hat{\Pi}(t)]_q + [C^b_w(t)]_q}. \tag{4.16}$$

Figure (4.3) describes the above equation as a block diagram.

There is a simple interpretation for (4.16). The smoothed high-resolution pixel at time t is the weighted average of the forward high-resolution estimate at time t ($[\hat{\underline{Z}}(t)]_q$) and the smoothed high-resolution pixel at time instant $t+1$ after motion compensation ($[\hat{\underline{Z}}^b_s(t)]_q$). In case there is high confidence in the $[\hat{\underline{Z}}(t)]_q$ (i.e., the value of $[\hat{\Pi}(t)]_q$ is small) the weight of

$[\underline{\hat{Z}}_s^b(t)]_q$ will be small. On the other hand, if there is high confidence in estimating the high-resolution pixel at time $t+1$ from a high-resolution pixel at time t after proper motion compensation (that is the value of $[C_w^b(t)]_q$ is small), it is reasonable to assume that the smoothed high-resolution pixel at time t can be estimated from a high-resolution pixel at time $t+1$ after proper motion compensation. Note that unlike the forward pass, estimation of high-resolution smoothed images do not depend on the computation of smoothed covariance update matrices as in Equations (4.14) and (4.15), and those can be ignored in the application.

The overall procedure using these update equations is outlined below in Algorithm 2.

```
• Task: Given $\{\underline{Y}(t)\}_{t\geq 1}$, estimate $\{\underline{Z}(t)\}_{t\geq 1}$ non-causally.

• First Pass: Assume that the causal algorithm has been applied, giving the sequence $\{\underline{\hat{Z}}(t), \hat{\Pi}(t)\}_{t=1}^{N}$.

• Initialization: Set $t = N$, choose $\underline{\hat{Z}}_s(t) = \underline{\hat{Z}}(t)$ and $\hat{\Pi}_s(t) = \hat{\Pi}(t)$.

• Update Process: Set $t \rightarrow t-1$ and apply

    1. Motion Compensation: Compute $\underline{\hat{Z}}_s^b(t) = F^T(t+1)\underline{\hat{Z}}_s(t+1)$
       and $\hat{\Pi}_s^b(t) = F^T(t+1)\hat{\Pi}_s(t+1)F(t+1)$.

    2. Update of the Covariance: Use Equation (4.15)
       to compute the update $\hat{\Pi}_s(t)$.

    3. Update of the Mean: Use Equation (4.16)
       to compute the update $\underline{\hat{Z}}_s(t)$.

• Repeat: Update process.
```

Algorithm 2: Smoothed dynamic Shift-and-Add algorithm.

4.3 Simultaneous Deblurring and Interpolation of Monochromatic Image Sequences

In the previous sections, we described two very fast image fusion algorithms, resulting in a blurry (possibly with some missing pixels) set of high-resolution images. In this section, we describe an effective method of deblurring and interpolation to produce the final high-quality reconstructed images. To perform robust deblurring and interpolation, we use the MAP cost function

$$\epsilon\left(\underline{X}(t)\right) = \|\Phi(t)(H\underline{X}(t) - \hat{\underline{Z}}(t))\|_2^2 + \lambda\Upsilon(\underline{X}(t)), \tag{4.17}$$

and define our desired solution as

$$\hat{\underline{X}}(t) = \underset{\underline{X}(t)}{\text{ArgMin}}\,\epsilon\left(\underline{X}(t)\right). \tag{4.18}$$

Here, the matrix $\Phi(t)$ is a diagonal matrix whose values are chosen in relation to our confidence in the measurements that contributed to make each element of$\hat{\underline{Z}}(t)$. These values have inverse relation to the corresponding elements in the matrix[3] $\hat{\Pi}(t)$. The regularization parameter, λ, is a scalar for properly weighting the first term (data fidelity cost) against the second term (regularization cost), and $\Upsilon(\underline{X})$ is the regularization cost function.

Following Chapter 2, we use the Bilateral Total Variation (B-TV) [4] as the regularization term. Therefore, for the case of monochromatic dynamic SR problem, the overall cost function is the summation of the data fidelity penalty term and the regularization penalty term

$$\begin{aligned}\hat{\underline{X}}(t)=&\underset{\underline{X}(t)}{\text{ArgMin}}\Bigg[\|\Phi(t)(H\underline{X}(t) - \hat{\underline{Z}}(t))\|_2^2 \\ &+\lambda\sum_{l,m=-P}^{P}\alpha^{|m|+|l|}\|\underline{X}(t) - S_x^l S_y^m \underline{X}(t)\|_1\Bigg]\ .\end{aligned} \tag{4.19}$$

As explained in Chapter 2, steepest descent optimization may be applied to minimize this cost function, which can be expressed as:

[3] Note that for the smoothed high-resolution estimation cases, $\hat{\underline{Z}}_s(t)$ and $\hat{\Pi}_s(t)$ substitute for $\hat{\underline{Z}}(t)$ and $\hat{\Pi}(t)$.

$$\begin{aligned}\underline{\hat{X}}_{n+1}(t) = & \underline{\hat{X}}_n(t) + \beta \Big\{ H^T \Phi^T(t)(H\underline{X}(t) - \underline{\hat{Z}}(t)) + \\ & \lambda \sum_{l,m=-P}^{P} \alpha^{|m|+|l|}[I - S_y^{-m} S_x^{-l}] sign \left(\underline{X}(t) - S_x^l \, S_y^m \underline{X}(t) \right) \Big\} \ . \end{aligned} \qquad (4.20)$$

4.4 Demosaicing and Deblurring of Color (Filtered) Image Sequences

Similar to what was described in Section 4.3, we deal with color sequences in a two step process of image fusion and simultaneous deblurring and interpolation for producing high quality color sequences from a collection of low-resolution color (filtered) images. Our computationally efficient MAP estimation method is motivated by the color image perception properties of the human visual system which is directly applicable to both color SR (given full RGB low-resolution frames), and the more general multi-frame demosaicing problems introduced earlier.

Figure 4.4 shows an overall block diagram of the dynamic SR process for mosaiced images (the feedback loops are eliminated to simplify the diagram). For the case of color SR, the first step involves nothing more than the application of the recursive image fusion algorithm separately on three different color bands. Image fusion of color filtered images is done quite similarly, where each single channel color filtered frame is treated as a sparsely sampled three-channel color image. The second step ("Deblur & Demosaic" block in Figure 4.4) is the enhancement step that removes blur, noise, and color artifacts from the Shift-and-Add sequence, and is based on minimizing a MAP cost function with several terms composing an overall cost function similar to $\epsilon\left(\underline{X}(t)\right)$ in (4.17).

The overall cost function $\epsilon\left(\underline{X}(t)\right)$ is the summation of these cost functions:

$$\underline{\hat{X}}(t) = \underset{\underline{X}(t)}{\text{ArgMin}} \left[J_0(\underline{X}(t)) + \lambda' J_1(\underline{X}(t)) + \lambda'' J_2(\underline{X}(t)) + \lambda''' J_3(\underline{X}(t)) \right] \ . \qquad (4.21)$$

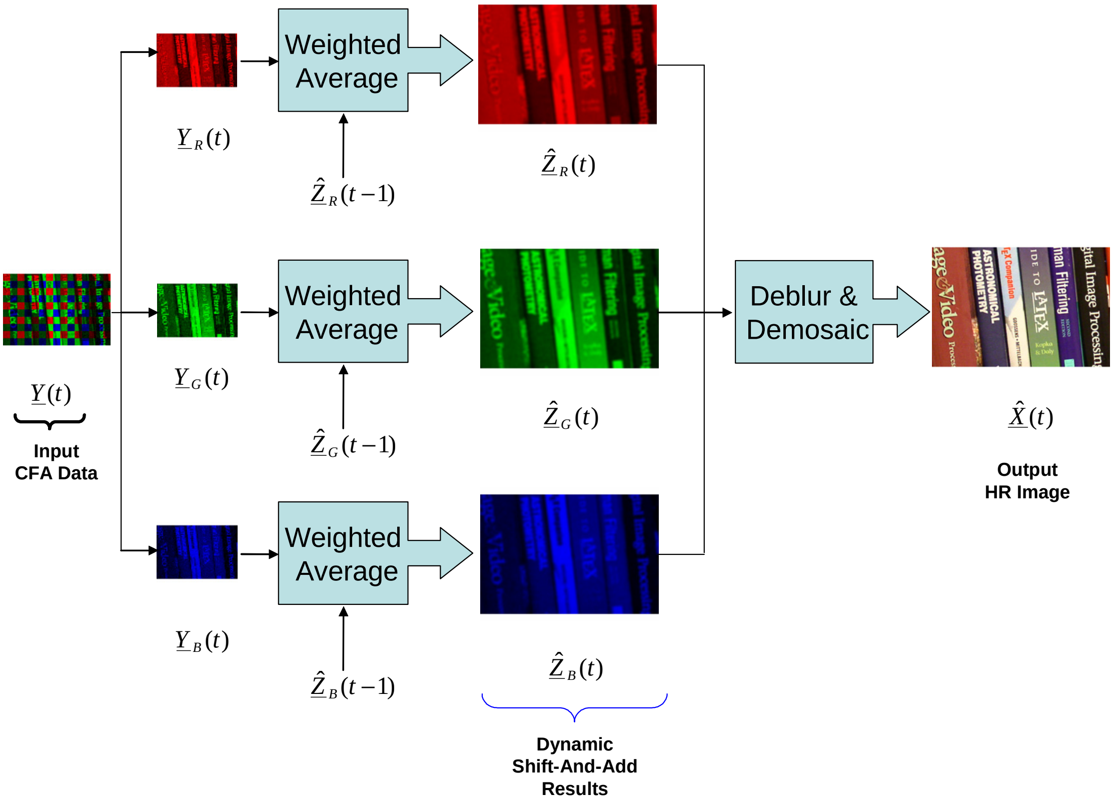


Figure 4.4: Block diagram representation of the overall dynamic SR process for color filtered images. The feedback loops are omitted to simplify the diagram. Note $\underline{\hat{Z}}_{i\in\{R,G,B\}}(t)$ represents the forward dynamic Shift-and-Add estimate studied in Section 4.2.2.

where the first term is replaced by

$$J_0(\underline{X}(t)) = \sum_{i=R,G,B} \|\Phi_i(t) \left(H\underline{\hat{X}}_i(t) - \underline{\hat{Z}}_i(t) \right) \|_2^2 \ , \tag{4.22}$$

and all other terms are similar to the ones formulated in Section 3.4.

Coordinate-wise steepest descent optimization may be applied to minimize this cost function. In the first step, the derivative of (4.21) with respect to one of the color bands is calculated, assuming the other two color bands are fixed. In the next steps, the derivative is computed with respect to the other color channels. The steepest descent iteration formulation for this cost function is shown in [75].

Note that $F(t)\underline{\hat{X}}(t-1)$ is a suitable candidate to initialize $\underline{\hat{X}}_0(t)$, since it follows the KF prediction of the state-vector updates. Therefore, as the deblurring-demosaicing step is the computationally expensive part of this algorithm, for all of these experiments we used the shifted version of deblurred image of $t-1$ as the initial estimate of the deblurred-demosaiced image at time instant t.

4.5 Experiments

Experiments on synthetic and real data sets are presented in this section. In the first experiment, we synthesized a sequence of low-resolution color-filtered images from a single color image of size 1200×1600 captured with a one-CCD OLYMPUS C-4000 digital camera. A 128×128 section of this image was blurred with a symmetric Gaussian low-pass filter of size 4×4 pixels with standard deviation equal to one (all three color bands). The resulting images were subsampled by the factor of four in each direction and further color filtered with Bayer pattern creating a 32×32 image. We added Gaussian noise to the resulting low-resolution frames to achieve SNR equal to 30dB. We consecutively shifted the 128×128 window on the original high-resolution image by one pixel in right, down, or up directions, and repeated the same image degradation process. In this fashion, we created a sequence of 250 frames.

Figures 4.5(a) & 4.5(e) show two sections of the high-resolution image. Figures

4.5(b) & 4.5(f) show frames #50 and #100 of the low-resolution sequence (for the sake of presentation each frame has been demosaiced following the method of [3]). We created a sequence of high-resolution fused images using the method described in Section 4.2.2 (factor of 4 resolution enhancement by forward Shift-and-Add method). Figures 4.5(c) & 4.5(g) show frames #50 and #100 of this sequence, where the missing values were filled in, using bilinear interpolation. Note that for the particular motion in this under-determined experiment, it is easy to show that less than $\frac{1}{3}$ of the pixel values in $\underline{\hat{Z}}(t)$ are determined by the Shift-and-Add process.

Later each frame was deblurred-demosaiced using the method described in Section 4.4. Figures 4.5(d) & 4.5(h) show frames #50 and #100 of this reconstructed sequence, where the color artifacts have been almost completely removed. Figure 4.6 shows similar experiments for frames #150 and #200, and Figure 4.7 shows the corresponding results for frame #250.

The PSNR values for this sequence are plotted in Figure 4.8. This plot shows that after the first few frames are processed, the quality of the reconstruction is stabilized for the remaining frames. The small distortions in the PSNR values of this sequence are due to the difference in color and high-frequency information of different frames. The corresponding parameters for this experiment (tuned by trial-and-error) were as follows: $\alpha = 0.9$, $\beta = 0.06$, $\lambda' = \lambda'' = 0.001$, and $\lambda''' = 10$. Fifteen iterations of steepest descent were used for this experiment.

Our next experiment was preformed on a real (already demosaiced) compressed image sequence courtesy of Adyoron Intelligent Systems Ltd., Tel Aviv, Israel. Two frames of this sequence (frames # 20 and #40) are shown in Figures 4.9(a) & 4.9(d). We created a sequence of high-resolution fused images (factor of 4 resolution enhancement) using the forward data fusion method described in Section 4.2.2 (Figures 4.9(b) & 4.9(e)). Later each frame in this sequence was deblurred using the method described in Section 4.4 (Figures 4.5(c) & 4.9(f)). The corresponding parameters for this experiment are as follows: $\alpha = 0.9$, $\beta = 0.1$, $\lambda' = \lambda'' = 0.005$, and $\lambda''' = 50$. Fifteen iterations of steepest descent were used for this experiment. The (unknown) camera PSF was assumed to be a 4×4 Gaussian kernel with standard deviation equal

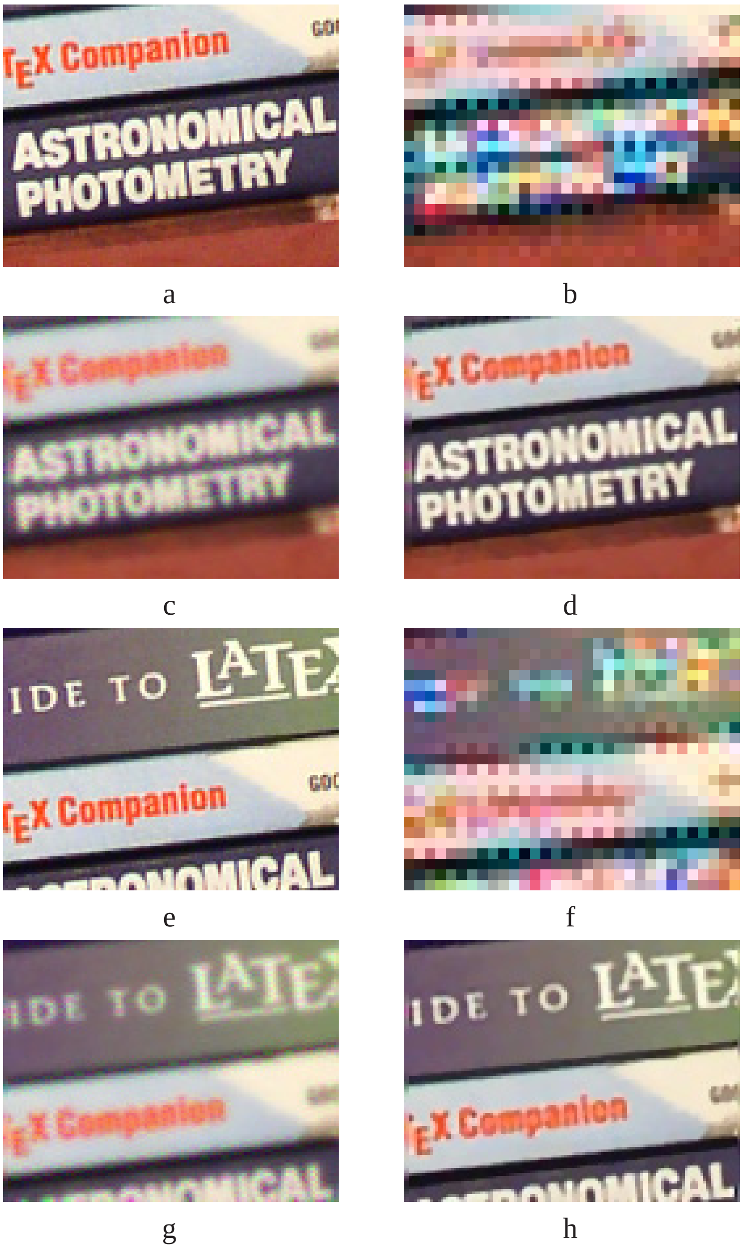


Figure 4.5: A sequence of 250 low-resolution color filtered images where recursively fused (Section 4.2), increasing their resolution by the factor of 4 in each direction. They were further deblurred and demosaiced (Section 4.4), resulting in images with much higher-quality than the input low-resolution frames. In (a) & (e) we see the ground-truth for frames #50 and #100 of size $[100 \times 128 \times 3]$, and (b) & (f) are the corresponding synthesized low-resolution frames of size $[25 \times 32 \times 3]$. In (c) & (g) we see the recursively fused high-resolution frames and (d) & (h) of size $[100 \times 128 \times 3]$ show the deblurred-demosaiced frames.

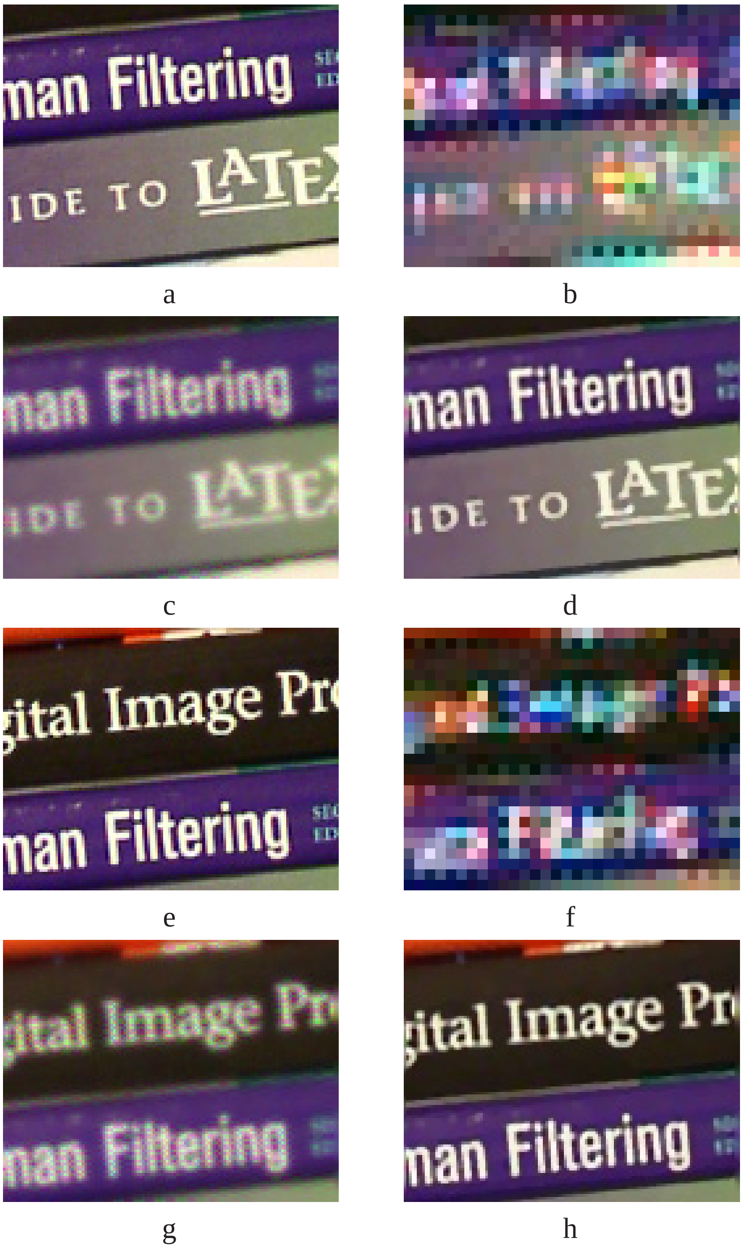
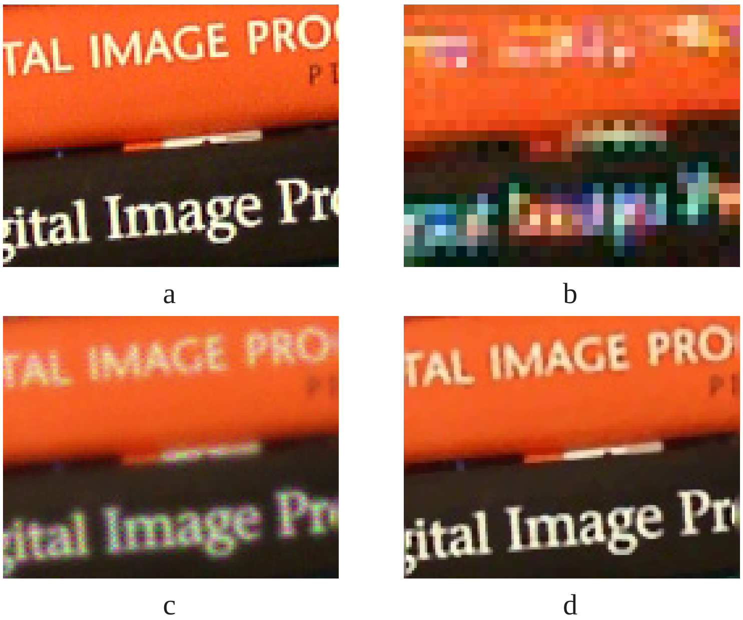


Figure 4.6: A sequence of 250 low-resolution color filtered images where recursively fused (Section 4.2), increasing their resolution by the factor of 4 in each direction. They were further deblurred and demosaiced (Section 4.4), resulting in images with much higher-quality than the input low-resolution frames. In (a) & (e) we see the ground-truth for frames #150 and #200 of size $[100 \times 128 \times 3]$, and (b) & (f) are the corresponding synthesized low-resolution frames of size $[25 \times 32 \times 3]$. In (c) & (g) we see the recursively fused high-resolution frames and (d) & (h) of size $[100 \times 128 \times 3]$ show the deblurred-demosaiced frames.



a b c d

Figure 4.7: A sequence of 250 low-resolution color filtered images where recursively fused (Section 4.2), increasing their resolution by the factor of 4 in each direction. They were further deblurred and demosaiced (Section 4.4), resulting in images with much higher-quality than the input low-resolution frames. In (a) we see the ground-truth for frame #250 of size $[100 \times 132 \times 3]$, and (b) is the corresponding synthesized low-resolution frame of size $[25 \times 32 \times 3]$. In (c) we see the recursively fused high-resolution frame and (d) of size $[100 \times 128 \times 3]$ show the deblurred-demosaiced frame.

to one. As the relative motion between these images approximately followed the translational model, we only needed to estimate the motion between the luminance components of these images [72]. We used the method described in [48] to compute the motion vectors. In the reconstructed images there are some effects of wrong motion estimation, seen as periodic teeth along the vertical bars. We assume that these errors correspond to the small deviations from the pure translational model.

In the third experiment, we used 74 uncompressed, raw CFA images from a video camera (based on Zoran 2MP CMOS Sensors) [4]. We applied the method of [3] to demosaic each of these low-resolution frames, individually. Figure 4.10(a) shows frame #1 of this sequence.

[4] We would like to thank Lior Zimet and Erez Galil from Zoran Corp. for providing the camera used to produce the raw CFA images of experiment 3 in Fig. 4.10.

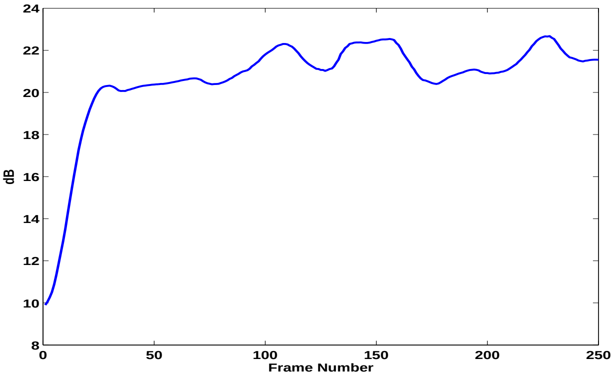


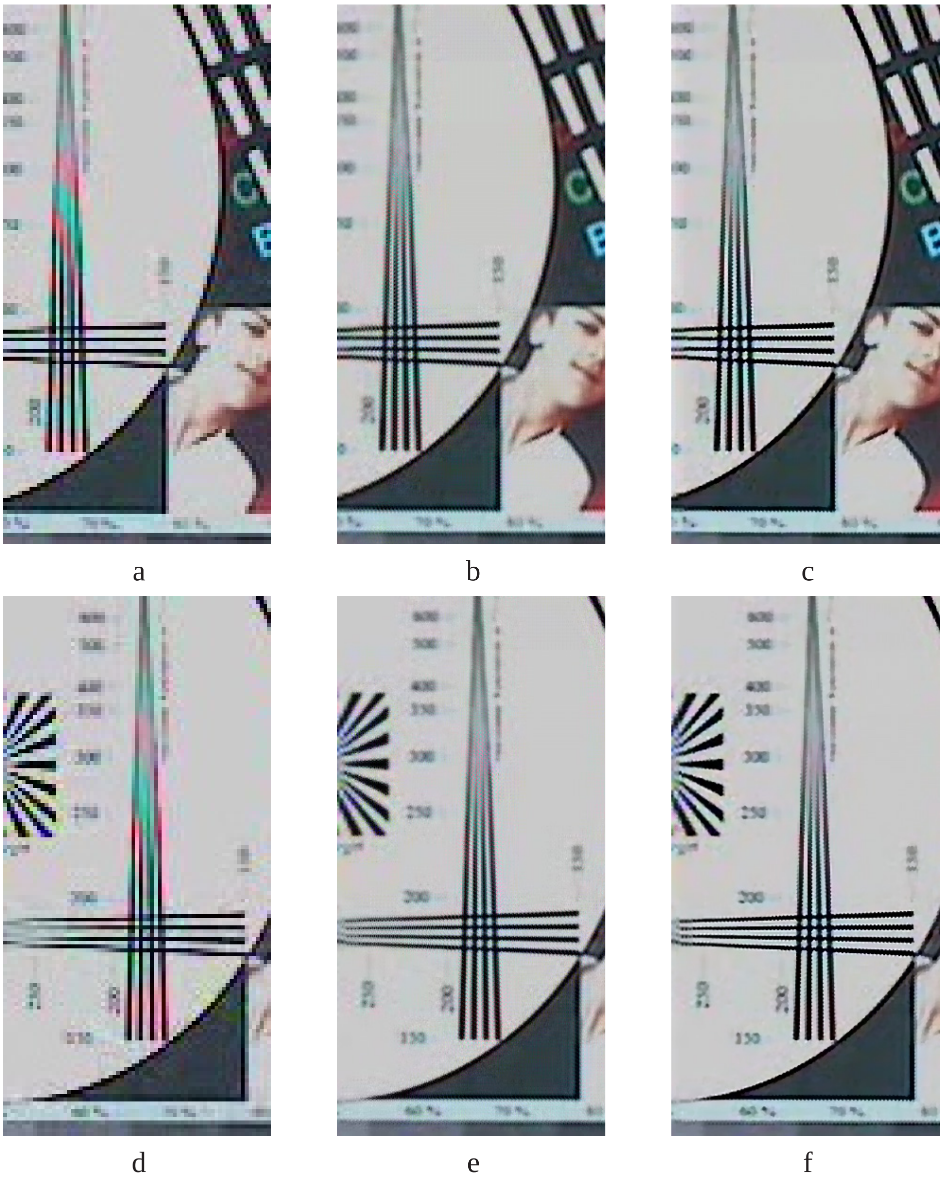
Figure 4.8: PSNR values in dB for the synthesized 250 frames sequence of the experiment in Figure 4.5.

To increase the spatial resolution by a factor of three, we applied the proposed forward data fusion method of Section 4.2.2 on the raw CFA data. Figure 4.10(b) shows the forward Shift-and-Add result. This frame were further deblurred-demosaiced by the method explained in Section 4.4 and the result is shown in Figures 4.10(c). To enhance the quality of reconstruction we applied the smoothing method of Section 4.2.3 to this sequence. Figure 4.10(d) shows the smoothed data fusion results for frames #1 (Smoothed Shift-and-Add). The deblurred-demosaiced result of applying the method explained in Section 4.4 is shown in Figure 4.10(e).

Figure 4.10(f) shows the frame #69 of this sequence, demosaiced by the method in [3]. Figure 4.10.g shows the result of applying the method of Section 4.2.3 to form the smoothed Shift-and-Add image. This frame is further deblurred-demosaiced by the method explained in Section 4.4 and the result is shown in Figure 4.10(h).

The parameters used for this experiment are as follows: $\beta = 0.04$, $\alpha = 0.9$, $\lambda^{'} = 0.001$, $\lambda^{''} = 50$, $\lambda^{'''} = .1$. The (unknown) camera PSF was assumed to be a tapered 5×5 disk PSF [5].

[5] MATLAB command fspecial('disk',2) creates such a blurring kernel.



a b c

d e f

Figure 4.9: A sequence of 60 real-world low-resolution compressed color frames (a & d of size $[141 \times 71 \times 3]$) are recursively fused (Section 4.2), increasing their resolution by the factor of four in each direction (b & e of size $[564 \times 284 \times 3]$). They were further deblurred (Section 4.4), resulting in images with much higher-quality than the input low-resolution frames (c & f).

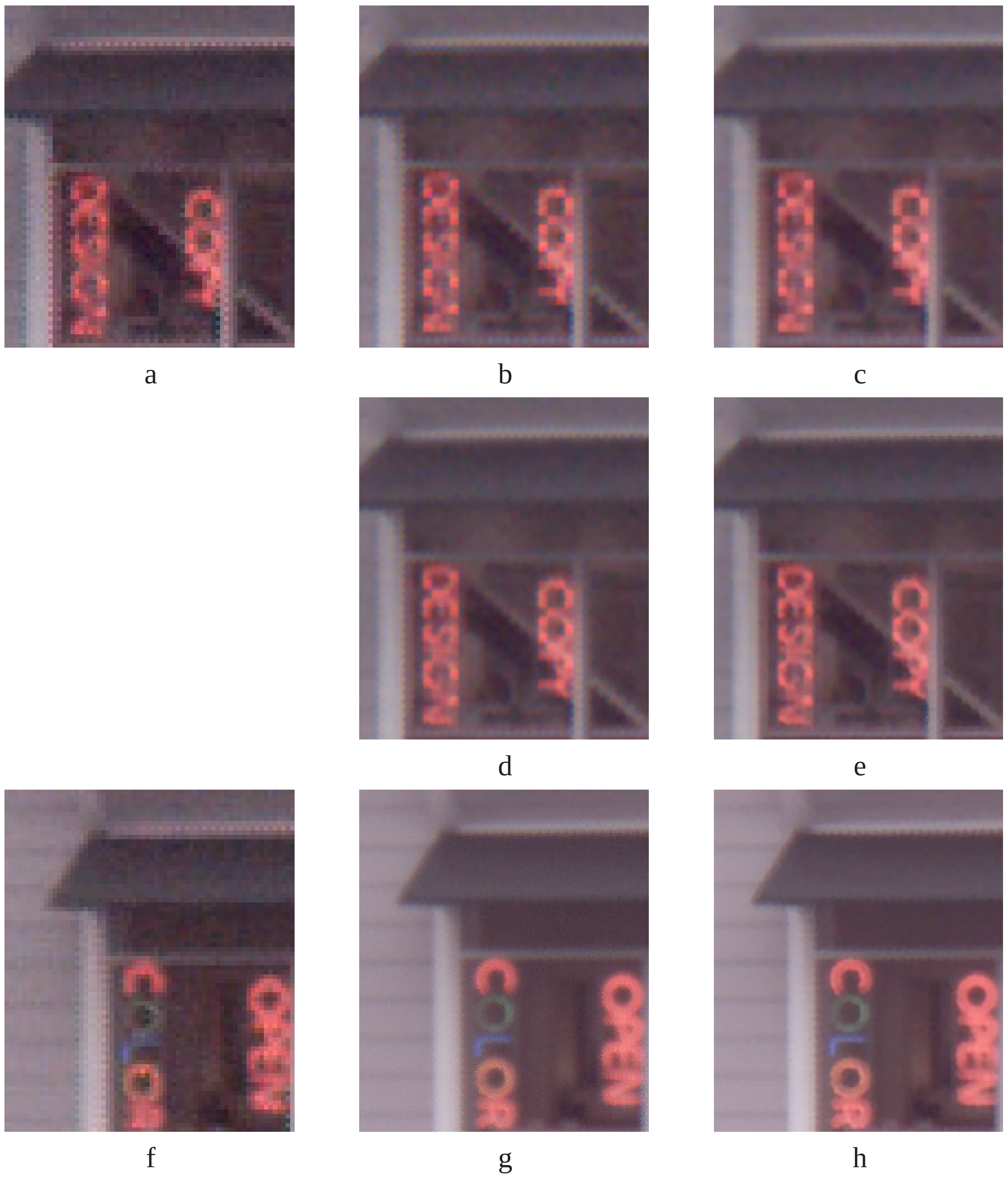


Figure 4.10: A sequence of 74 real-world low-resolution uncompressed color filtered frames of size $[76 \times 65 \times 3]$ (a & f show frames #1 and #69, respectively) are recursively fused (Forward data fusion method of Section 4.2.2), increasing their resolution by the factor of three in each direction (b & g of size $[228 \times 195 \times 3]$). They were further deblurred (Section 4.4), resulting in images with much higher-quality than the input low-resolution frames (c & h). The smoothed data fusion method of Section 4.2.3 further improves the quality of reconstruction. The smoothed Shift-and-Add result for frame #1 is shown in (d). This image was further deblurred-demosaiced (Section 4.4) and the result is shown in (e).

4.6 Summary and Discussion

In this chapter, we presented algorithms to enhance the quality of a set of noisy, blurred, and possibly color filtered images to produce a set of monochromatic or color high-resolution images with less noise, aliasing, and blur effects. We used MAP estimation technique to derive a hybrid method of dynamic SR and multi-frame demosaicing. Our method is also applicable to the case of color SR.

For the case of translational motion and common space-invariant motion we justified a two-step algorithm. In the first step, we used the KF framework for fusing low-resolution images recursively in a fast and memory efficient way. In the second step, while deblurring and interpolating the missing values, we reduced luminance and color artifacts by using appropriate penalty terms. These terms were based on our prior knowledge of the statistics of natural images and the properties of the human visual system. All matrix-vector operations in the proposed method are implemented as simple image operators.

Chapter 5

Constrained, Globally Optimal Multi-Frame Motion Estimation

5.1 Introduction

So far in this thesis, we assumed that the system matrix M in (1.1), is known before hand or given from a separate estimation process. The system matrix is usually modeled as a combination of three separate matrices; namely warp, blur, and down-sampling (2.4). Of these three terms, accurate estimation of the warping matrix is of greatest importance [4]. Note that, errors in motion estimation might even result in reconstruction of HR frames which have lower quality than the input LR frames. Besides, motion estimation with subpixel accuracy is of great importance to many other image processing and computer vision applications, such as mosaicing [47].

Numerous image registration techniques have been developed throughout the years [80]. Of these, optical flow [81] [82], and correlation-based methods [83] are among the most popular. These methods are mainly developed to estimate the relative motion between *a pair* of frames. For cases where several images are to be registered with respect to each other (e.g. super-resolution applications), two simple strategies are commonly used. The first is to regis-

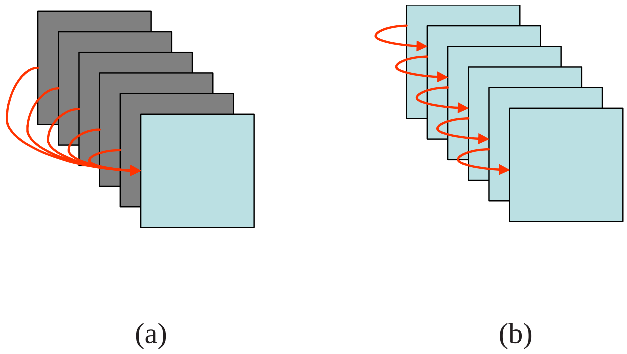


Figure 5.1: Common strategies used for registering frames of a video sequence. (a) Fixed reference ("anchored") estimation. (b) Pairwise ("progressive") estimation.

ter all frames with respect to a single reference frame [4]. This may be called the *anchoring* approach, as illustrated in Figure 5.1(a). The choice of a reference or anchor frame is rather arbitrary, and can have a severe effect on the overall accuracy of the resulting estimates. This caveat aside, overall, this strategy is effective in cases where the camera motion is small and random (e.g. small vibrations of a gazing camera).

The other popular strategy is the *progressive* registration method [21] [84], where images in the sequence are registered in pairs, with one image in each pair acting as the reference frame. For instance, taking a causal view with increasing index denoting time, the i^{th} frame of the sequence is registered with respect to the $(i+1)^{\text{th}}$ frame and the $(i+1)^{\text{th}}$ frame is registered with respect to the $(i+2)^{\text{th}}$ frame, and so on, as illustrated in Figure 5.1(b). The motion between an arbitrary pair of frames is computed as the combined motion of the above incremental estimates. This method works best when the camera motion is smooth. However, in this method, the registration error between two "nearby" frames is accumulated and propagated when such values are used to compute motion between "far away" frames.

Neither of the above approaches take advantage of the important prior information available for the multi-frame motion estimation problem. This prior information constrains the estimated motion vector fields between any pair of frames to lie in a space whose geometry and structure, as we shall see in the next section, is conveniently described.

In this chapter, we study such priors and propose an optimal method for exploiting them, to achieve very accurate estimation of the relative motion in a sequence. This chapter is organized as follows. Section 5.2 introduces the consistency constraints in an image sequence and reviews the previous work on this subject. Section 5.3 describes the main contribution of this chapter, which is an optimal framework for exploiting these consistency constraints. Using this framework, we introduce a highly accurate robust multi-frame motion estimation method, which is resilient to outliers in an image sequence. experiments based on both real and synthetic data sequences are presented in Section 5.4, and Section 5.5 concludes this chapter.

5.2 Constrained Motion Estimation

To begin, let us define $\mathbf{F_{i,j}}$ as the operator which maps (registers) frames indexed i and j as follows:

$$\underline{Y}_i = \mathbf{F_{i,j}}\{\underline{Y}_j\},$$

where $\underline{Y}_i$ and $\underline{Y}_j$ are the lexicographic reordered vector representations of frames i and j.

Now given a sequence of N frames, precisely $N(N-1)$ such operators can be considered. Regardless of considerations related to noise, sampling, and the finite dimensions of the data, there are inherent intuitive relationships between these pair-wise registration operators. In particular, the first condition dictates that the operator describing the motion between any pair of frames must be the composition of the operators between two other pairs of frames. More specifically, as illustrated in Figure 5.2(a), taking any triplet of frames i, j, and k, we have the first motion consistency condition as:

$$\forall i, j, k \in \{1, ..., N\}, \quad \mathbf{F_{i,k}} = \mathbf{F_{i,j}} \circ \mathbf{F_{j,k}}. \tag{5.1}$$

The second rather obvious (but hardly ever used) consistency condition states that the composition of the operator mapping frame i to j with the operator mapping frame j to i should yield the identity operator. This is illustrated in Figure 5.2(b). Put another way,

$$\forall i, j \in \{1, ..., N\}, \quad \mathbf{F_{j,i}} = \mathbf{F_{i,j}^{-1}}. \tag{5.2}$$

These natural conditions define an algebraic group structure (a Lie algebra) in which the operators reside. Therefore, any estimation of motion between frames of a $(N \gg 2)$ image sequence could take these conditions into account. In particular, the optimal motion estimation strategy can be described as an estimation problem over a group structure, which has been studied before in other contexts [85].

The above properties describe what is known as the Jacobi condition, and the skew anti-symmetry relations [86]. For some practical motion models (e.g. constant motion or the affine model), the relevant operators could be further simplified. For example, in the case of translational (constant) motion, the above conditions can be described by simple linear equations relating the (single) motion vectors between the frames:

$$\forall i, j, k \in \{1, ..., N\}, \quad \underline{\delta}_{i,k} = \underline{\delta}_{i,j} + \underline{\delta}_{j,k}, \tag{5.3}$$

where $\underline{\delta}_{i,j}$ is the motion vector between the frames i and j. Note that $\underline{\delta}_{i,i} = \underline{0}$, and therefore the skew anti-symmetry condition is represented by (5.3), when $k = i$.

For the sake of completeness, we should note that the above ideas have been already studied to some extent in the computer vision community. In particular, the Bundle Adjustment (BA) [87] technique is a general, yet computationally expensive method for producing a jointly optimal 3D structure and viewing parameters, which bares close resemblance to what is proposed here. It is important to note that BA is not intended for motion estimation in 2-D images, and does not specifically take the algebraic group structure into account. Instead, it relies on an iterative method, which is largely based on the motivating 3-D application. On another front, to solve mosaicing problems, [88] adapted the BA method to a 2-D framework, where the estimated motion vectors are refined in a feedback loop, penalizing the global inconsistencies between frames. Also, the importance of consistent motion estimation for the Super-Resolution problem is discussed in [82].

In [89] the Group structure is directly exploited to define a one step multi-frame motion estimation method, where the motion model is limited to rotation and translation. The very recent approach in [86] exploits the Lie Group structure indirectly. The motions are estimated

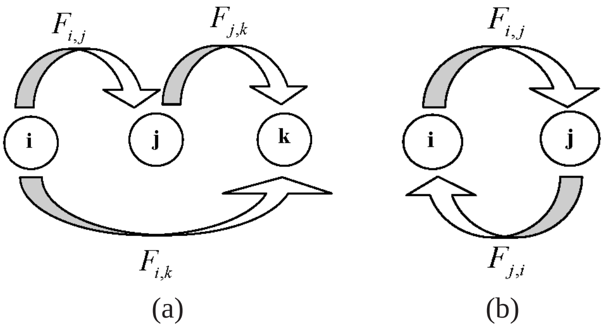


Figure 5.2: The consistent flow properties: (a) Jacobi Identity and (b) Skew Anti-Symmetry.

in an unconstrained framework, then "projected" to the set of valid motions by what the author calls Lie-algebraic averaging. While the framework of this approach is close to what we suggest, the algorithm presented there is suboptimal in that it uses the constraints only as a mechanism for post-processing already-estimated motion fields, resulting in a suboptimal overall procedure. Finally, in a similar way, another recent paper, [90], computes the motion vectors between a new frame and a set of frames for which relative motion vectors has been previously computed. Then, the motion vectors computed for the new image are used to refine the pairwise estimated motion of other frames. This two-step algorithm is iterated until convergence.

The framework we propose in this chapter unifies the earlier approaches and presents an optimal framework where the constraints are used directly in the solution of the problem, and not simply as a space onto which the estimates are projected.

5.3 Precise Estimation of Translational Motion with Constraints

We now describe our proposed methodology, and compare it against two other competing approaches. To simplify the notation, we define the vectors $\underline{Y}$, $\underline{\delta}$, and $\underline{\delta}(i)$ as follows:

$$\underline{Y} = \begin{bmatrix} \underline{Y}(1) \\ \underline{Y}(2) \\ \vdots \\ \underline{Y}(N) \end{bmatrix}, \underline{\delta} = \begin{bmatrix} \underline{\delta}(1) \\ \underline{\delta}(2) \\ \vdots \\ \underline{\delta}(N) \end{bmatrix}, \underline{\delta}(i) = \begin{bmatrix} \underline{\delta}_{i,1} \\ \vdots \\ \underline{\delta}_{i,j(i\neq j)} \\ \vdots \\ \underline{\delta}_{i,N} \end{bmatrix}, \tag{5.4}$$

where $\underline{Y}(i)$ is the i^{th} image in this sequence rearranged in the lexicographic order. The vector $\underline{\delta}(i)$ contains the set of motion vector fields computed with respect to the reference frame i.

5.3.1 Optimal Constrained Multi-Frame Registration

In a general setting, the optimal solution to the multi-frame registration problem can be obtained by minimizing the following cost function:

$$\widehat{\underline{\delta}} = \underset{\underline{\delta}}{\text{ArgMin}}\, \rho(\underline{Y}, \underline{\delta}) \text{ such that } \Upsilon(\underline{\delta}) = 0, \tag{5.5}$$

where ρ represents a motion-related cost function (e.g. penalizing deviation from brightness constancy constraint, or a phase-based penalty), and Υ captures the constraints discussed earlier.

To get a feeling for this general formulation, we address the translational motion case, (the consistency conditions for the affine case are described in the Appendix F), with ρ representing the Optical Flow model

$$\rho(\underline{Y}, \underline{\delta}) = \sum_{\underbrace{i,j=1}_{i\neq j}}^{N} \|\underline{Y}_i^{(x)} \delta_{i,j}^{(x)} + \underline{Y}_i^{(y)} \delta_{i,j}^{(y)} + \underline{Y}_{i,j}^{(t)})\|_2^2, \tag{5.6}$$

where $\underline{Y}_i^{(x)}$ and $\underline{Y}_i^{(y)}$ are the spatial derivatives (in x and y directions) of the i^{th} frame, and $\underline{Y}_{i,j}^{(t)}$ is the temporal derivative (e.g., the difference between frames i and j). Here the motion vector field $\underline{\delta}_{i,j}$ is spatially constant, and it can be represented by the scalar components $\delta_{i,j}^{(x)}$ and $\delta_{i,j}^{(y)}$ in x and y directions, respectively, and for $1 \leq i, j \leq N$. Using this, the translational consistency condition as in Equation (5.3) is then formulated as

$$\Upsilon(\underline{\delta}) : \quad C\underline{\delta} = 0, \tag{5.7}$$

where the unknown motion vector $\underline{\delta}$ has all the $2N(N-1)$ entries $\delta_{i,j}^{(x)}$ and $\delta_{i,j}^{(y)}$ stacked into a vector. The constraint matrix C is of size $[2(N-1)^2 \ \times \ 2N(N-1)]$. Each row in C has only two or three non-zero (± 1) elements representing the skew anti-symmetry and Jacobi identity conditions in (5.3), respectively. The defined problem has a quadratic programming structure, and it can be solved using accessible optimization algorithms.

5.3.2 Two-Step Projective Multi-Frame Registration

Two-Step Projective Sub-Optimal Multi-Frame Registration

As a comparison to our proposal, we discuss a two-step approach that is in spirit similar to what is done in [86]. In this method, for a sequence of N frames, in the first step all $N(N-1)$ possible pairwise motion vector fields ($\underline{\delta}_{i,j}$) are estimated. Note that the pairwise motion vector fields are individually estimated by optimizing the following (unconstrained) cost function

$$\widehat{\underline{\delta}}_{i,j} = \underset{\underline{\delta}_{i,j}}{\text{ArgMin}} \, \rho(\underline{Y}_{i,j}, \underline{\delta}_{i,j}),$$

where $\rho(\underline{Y}_{i,j}, \underline{\delta}_{i,j})$ may represent any motion estimation cost function.

In the second step, these motion vectors are projected onto a consistent set of $N(N-1)$ pairwise motion vectors. For the case of translational motion model, with the consistency condition in (5.7), the projection of the motion vector fields onto the constraint space is computed as

$$\underline{\delta}_p = \Omega_p \widehat{\underline{\delta}} = (I - C[C^T C]^{-1} C^T) \widehat{\underline{\delta}}, \tag{5.8}$$

where $\Omega_p = I - C[C^T C]^{-1} C^T$ is the projection matrix. Such a two step projection method (as in [86]) is not optimal and would be expected to result in inferior estimates as compared to the solution of the method posed in Equation (5.5).

Two-Step Projective Optimal Multi-Frame Registration

In cases where the motion-related cost function ρ is manifested by the L_2 norm (e.g. (5.6)), it is also possible to design an optimal two step projective method, which results in estimates equal to the ones given by the optimal one-step method of Section 5.3.1. Note that the optical flow equation of (5.6) can be represented as

$$\rho(\underline{Y}, \underline{\delta}) = \|\underline{Y}^{(z)}\underline{\delta} + \underline{Y}^{(t)}\|_2^2, \tag{5.9}$$

where

$$\underline{Y}^{(z)} = \begin{bmatrix} \underline{Y}^{(x)}(1) \\ \underline{Y}^{(y)}(1) \\ \underline{Y}^{(x)}(2) \\ \underline{Y}^{(y)}(2) \\ \vdots \\ \underline{Y}^{(x)}(N) \\ \underline{Y}^{(y)}(N) \end{bmatrix}, \quad \underline{Y}^{(t)} = \begin{bmatrix} \underline{Y}^{(t)}(i,1) \\ \vdots \\ \underline{Y}^{(t)}(i,j)_{(i \neq j)} \\ \vdots \\ \underline{Y}^{(t)}(i,N) \end{bmatrix}. \tag{5.10}$$

Then, utilizing the Lagrangian multipliers method [91], the optimal constrained estimate is given by

$$\underline{\delta}_{op} = \Omega_{op}\widehat{\underline{\delta}} = \left(I - [\underline{Y}^{T(z)}\underline{Y}^{(z)}]^{-1} C [C^T [\underline{Y}^{T(z)}\underline{Y}^{(z)}]^{-1} C]^{-1} C^T\right)\widehat{\underline{\delta}}, \tag{5.11}$$

where Ω_{op} is the optimal projection matrix. Of course, storing (putting aside inverting) the matrix $[\underline{Y}^{T(z)}\underline{Y}^{(z)}]$ is computationally cumbersome, and therefore for the cases involving relatively large number of frames (or large images) the method of Section 5.3.1 is preferable.

5.3.3 Robust Multi-Frame Registration

In many real video sequences, the practical scenarios are not well-modeled by temporally stationary noise statistics, and abrupt changes or occlusions may introduce significant outliers into the data. Note that even the presence of very small amount of outliers, which may be unavoidable (e.g. the bordering pixel effects), heavily affects the motion estimation accuracy. In such cases, it is prudent to modify the above approaches in two ways. First, one may replace the hard constraints developed above with soft ones, by introducing them as Bayesian priors which will *penalize* rather than *constrain* the optimization problem. Second, we may want to introduce alternative norms to the standard 2-norm for both the error term and the constraint in (5.5). Incorporating both modifications, one can consider optimizing a modified cost function which includes a term representing the "soft" version of the constraints as

$$\widehat{\underline{\delta}} = \underset{\underline{\delta}}{\text{ArgMin}}\, \rho_r(\underline{Y}, \underline{\delta}) + \lambda \Upsilon(\underline{\delta}), \tag{5.12}$$

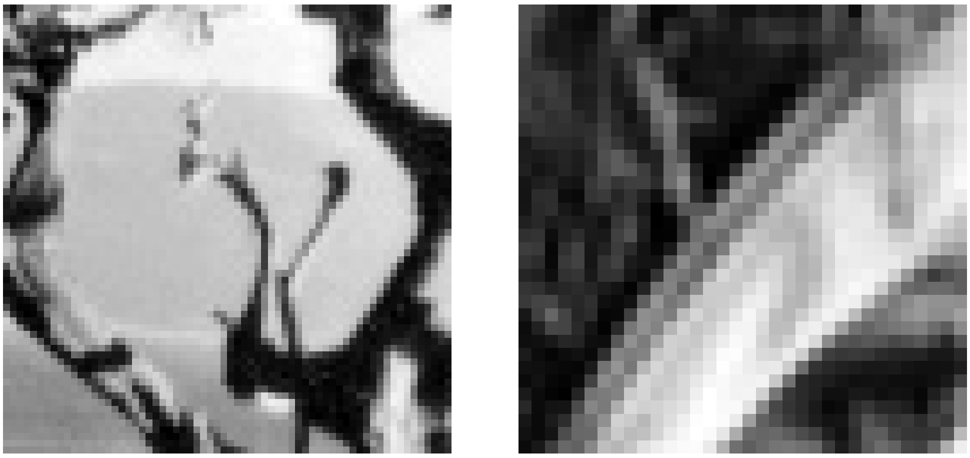
where λ represents the strength of the regularizing term. The functions ρ and Υ may use robust measures, such as the 1-norm. For instance, to deal with outliers directly, one might use

$$\rho_r(\underline{Y}, \underline{\delta}) = \sum_{\underbrace{i,j=1}_{i \neq j}}^{N} \|\underline{Y}_i^{(x)} \delta_{i,j}^{(x)} + \underline{Y}_i^{(y)} \delta_{i,j}^{(y)} + \underline{Y}_{i,j}^{(t)})\|_1. \tag{5.13}$$

The use of such robust error terms together with the hard constraint cost function of (5.5) often suffices to enhance the estimator performance. Note that unlike the L_2 norm which reduces the estimation error by an implicit averaging of the estimates, the robust L_1 norm implements a median estimator [40], which effectively picks the most reliable estimated motion vector for each pair of frames. The experiments in the next section justify this claim.

5.4 Experiments

A simulated experiment was conducted by registering 5 frames of size $[65 \times 65]$. For these frames we have the correct translational motion vectors in hand. One of these frames is



a: Experiment 1

b:Experiment 2

Figure 5.3: One of the input frames used in the first and the second experiments (simulated motion).

shown in Figure 5.3(a).

The mean square errors (MSEs) of the computed motion vectors (against the true motion vectors) with the single reference approach (Section 5.1), suboptimal projective (Section 5.3.2), the L_2 constrained (Section 5.3.1), and the L_1 norm with hard constraints (Section 5.3.3) methods are compared in Figure 5.4. Each point in this graphs shows the average of 100 different realizations of additive noise (Monte Carlo simulation) for different SNRs.

The second simulated experiment was conducted by registering 7 frames of size $[39 \times 39]$. One of these frames is shown in Figure 5.3(b). We repeated the previous experiment on this data set (with 30 Monte Carlo iterations for different SNRs) and compared the performance of different methods in Figure 5.5.

A real experiment was also conducted aligning 27 color-filtered[1] low-resolution (LR) images. One of these LR frames after demosaicing 3 is shown in Figure 5.6(a). The method of [75] was used to construct a HR image, by registering these images on a finer grid (resolution enhancement factor of three in x and y directions). We used the method described in [48] to compute the motion vectors in an “anchored” fashion (Section 5.1). Figure 5.6(b) shows the HR

[1] We would like to thank Eyal Gordon from the Technion-Israel Institute of Technology for helping us capture the raw CFA images used in the Figure 5.6 experiment.

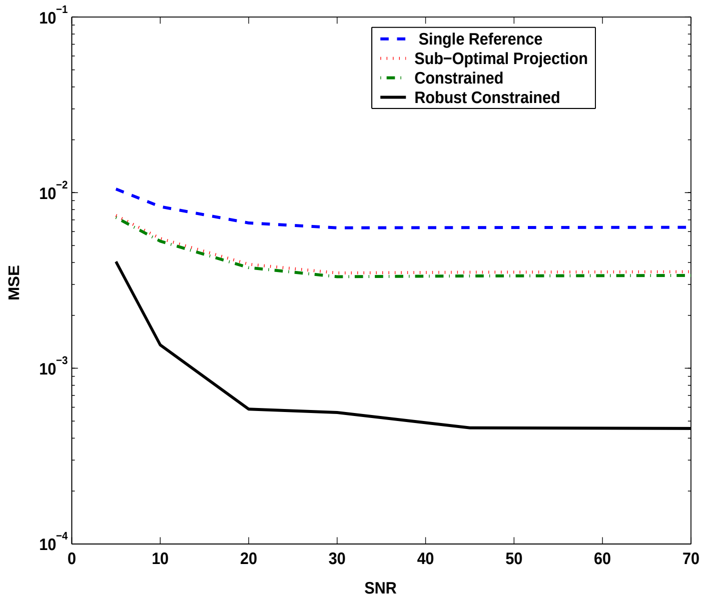


Figure 5.4: MSE comparison of different registration methods in the first simulated experiment, using the image in Figure 5.3(a).

reconstruction using this method with clear mis-registration errors. The result of applying the two step multi-frame projective image registration of Section 5.3.2 is shown in Figure 5.6(c). Some mis-registration errors are still visible in this result. Finally, the result of applying the optimal multi-frame registration method (Section 5.3.1) is shown in Figure 5.6(d), with almost no visible mis-registration error.

5.5 Summary and Discussion

In this chapter we studied several multi-frame motion estimation methods, focusing on the methods that exploit the consistency conditions. As an alternative to existing methods, we proposed a general framework to optimally benefit from these constraints. Such a framework is flexible, and is applicable to more general motion models. Based on this framework, we proposed a highly accurate multi-frame motion estimation method which is robust to the outliers in image sequences. This robust method, which minimizes an L_1 norm cost func-

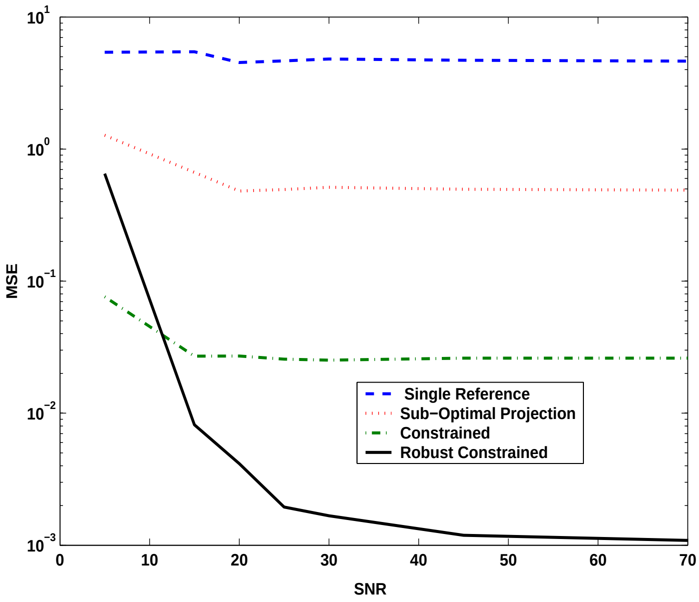
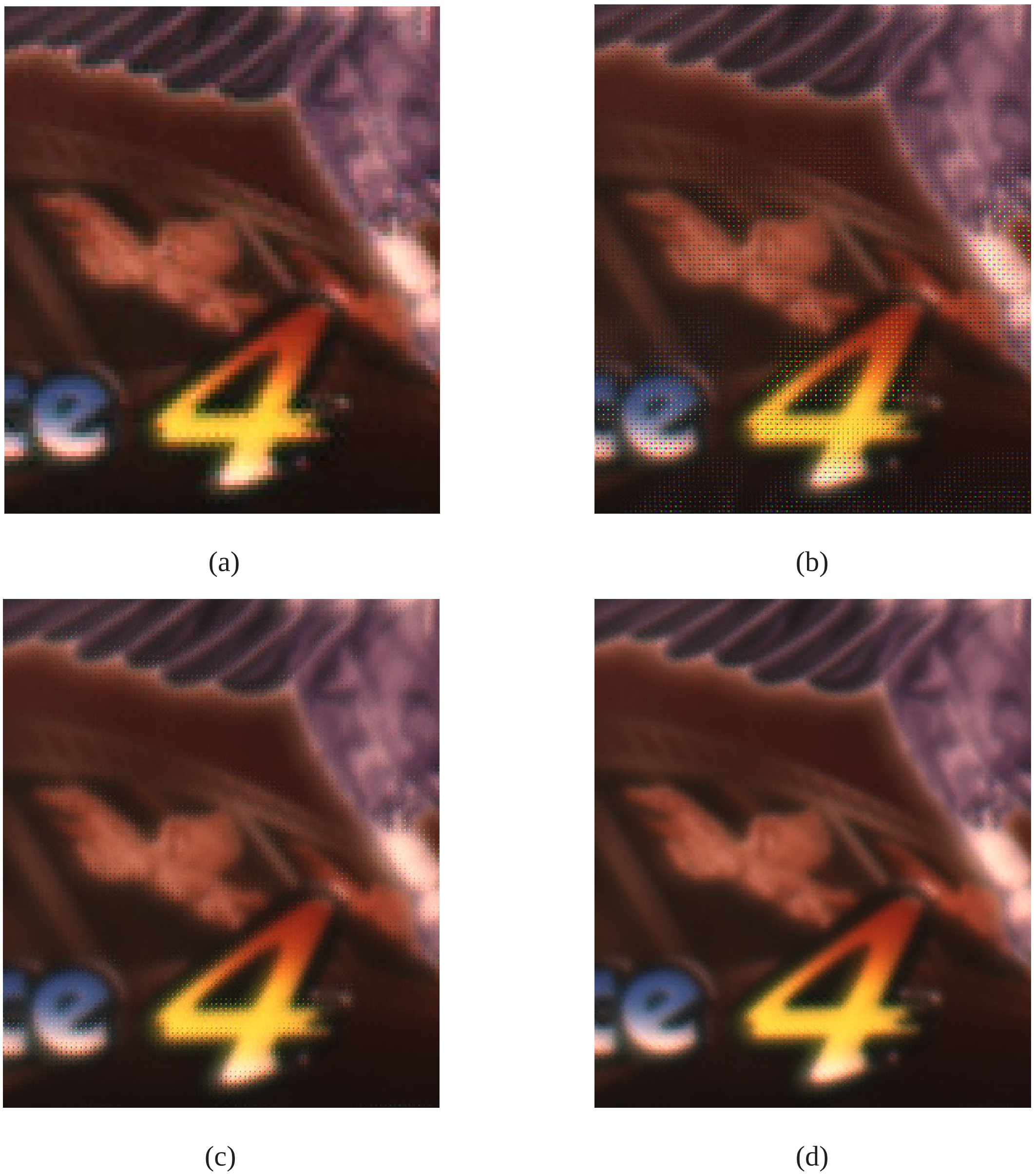


Figure 5.5: MSE comparison of different registration methods in the second simulated experiment, using the image in Figure 5.3(b).

tion, often provides more accurate estimation than the common least square approaches. Our experiments show that the high accuracy and reliability of the proposed multi-frame motion estimation method is especially useful for the multi-frame super-resolution applications in which very accurate motion estimation is essential for effective image reconstruction.



(a)

(b)

(c)

(d)

Figure 5.6: Experimental registration results for a real sequence. (a) One input LR frame after demosaicing.(b) Single Reference HR registration. (c) Projective HR registration. (d) Optimal HR registration.

Chapter 6

Conclusion and Future work

This chapter summarizes the contributions of this thesis in the field of multi-frame image fusion and enhancement. We also detail several open questions related to this thesis as well as map out future research direction.

6.1 Contributions

In this thesis, we proposed a general Maximum A-Posteriori framework for solving multi-frame image fusion problems. This framework helped us construct a well-defined description of several aspects of this problem from an estimation theoretic point of view, allowing us to make fundamental contributions to both the methodology and the science of image fusion, reconstruction, and enhancement.

- In Chapter 2, we described a general theory regarding robust estimation of a high-quality image from a set of low-quality images. In particular, we focused on the grayscale super-resolution problem of constructing a high-resolution image from a set of low-resolution noisy and blurred images. We showed that the reconstruction errors induced due to motion estimation follow the Laplacian model and often play a critical role in the overall performance of any multi-frame super-resolution method. Unlike classic SR methods, we advocated the use of the L_1 norm to fuse low-resolution images which significantly reduced the effects of the outliers. To further enhance the quality of reconstruction, we introduced an adaptive regularization method called Bilateral Total Variation which resulted in high-resolution images with sharp edges. Furthermore, for the case of translational motion, and common space invariant blur, we justified an effective two step "Shift-and-Add" SR method which is very fast to implement.

- In Chapter 3, we showed that the multi-frame super-resolution of color images and a common technique in commercial digital cameras called demosaicing are the special cases of a general problem called multi-frame demosaicing. Based on the general MAP estimation framework of the Chapter 2, we addressed these problems optimally under a unified context. By minimizing a multi-term cost function, we proposed a fast and robust hybrid method of super-resolution and demosaicing. The L_1 norm was used for measuring the difference between the projected estimate of the high-resolution image and each low-resolution image, removing outliers in the data and errors due to possibly inaccurate motion estimation. Bilateral Total Variation regularization was used for spatially regularizing the luminance component, resulting in sharp edges and forcing interpolation along the edges and not across them. Simultaneously, Tikhonov regularization was used to smooth the chrominance components. Finally, an additional regularization term was used to force similar edge location and orientation in different color channels. We showed that the minimization of the total cost function is relatively easy and fast.

- In Chapter 4, we addressed the dynamic super-resolution problem of reconstructing a high-quality set of monochromatic or color super-resolved images from low-quality monochromatic, color, or mosaiced frames. Our approach includes a joint method for simultaneous SR, deblurring, and demosaicing, this way taking into account practical color measurements encountered in video sequences. For the case of translational motion and common space-invariant blur, the proposed method was based on a very fast and memory efficient approximation of the Kalman Filter, and the Shift-and-Add process studied in the previous chapters. We presented two closely related implementations of our proposed method. The first technique is an extremely fast and memory efficient *causal* method suitable for realtime (on-line processing) applications. The second more accurate *non-causal* method, which is suitable for off-line processing applications, is still much more time and memory efficient than the corresponding static super-resolution methods.

- In Chapter 5, we studied several multi-frame motion estimation methods, focusing on the methods that exploit the motion consistency conditions. As an alternative to existing methods, we proposed a general framework to optimally benefit from these constraints. Such framework is flexible, and is applicable to more general motion models. Based on this framework, we proposed a highly accurate multi-frame motion estimation method which is robust to outliers in image sequences. This robust method, which minimizes an L_1 norm cost function, often provides more accurate estimation than the common least square approaches. Our experiments showed that the high accuracy and reliability of the proposed multi-frame motion estimation method is especially useful for the multi-frame super-resolution/demosaicing applications in which very accurate motion estimation is essential for effective image reconstruction.

- Based on the material presented in this thesis, we have developed a Matlab-based software package, called "MDSP Resolution Enhancement Software Package". The main objective of this software tool is the implementation and comparison of several super-resolution, demosaicing, and motion estimation techniques. In particular, the techniques described in this thesis, and several references therein are included.

  Some specific features of the software package are:

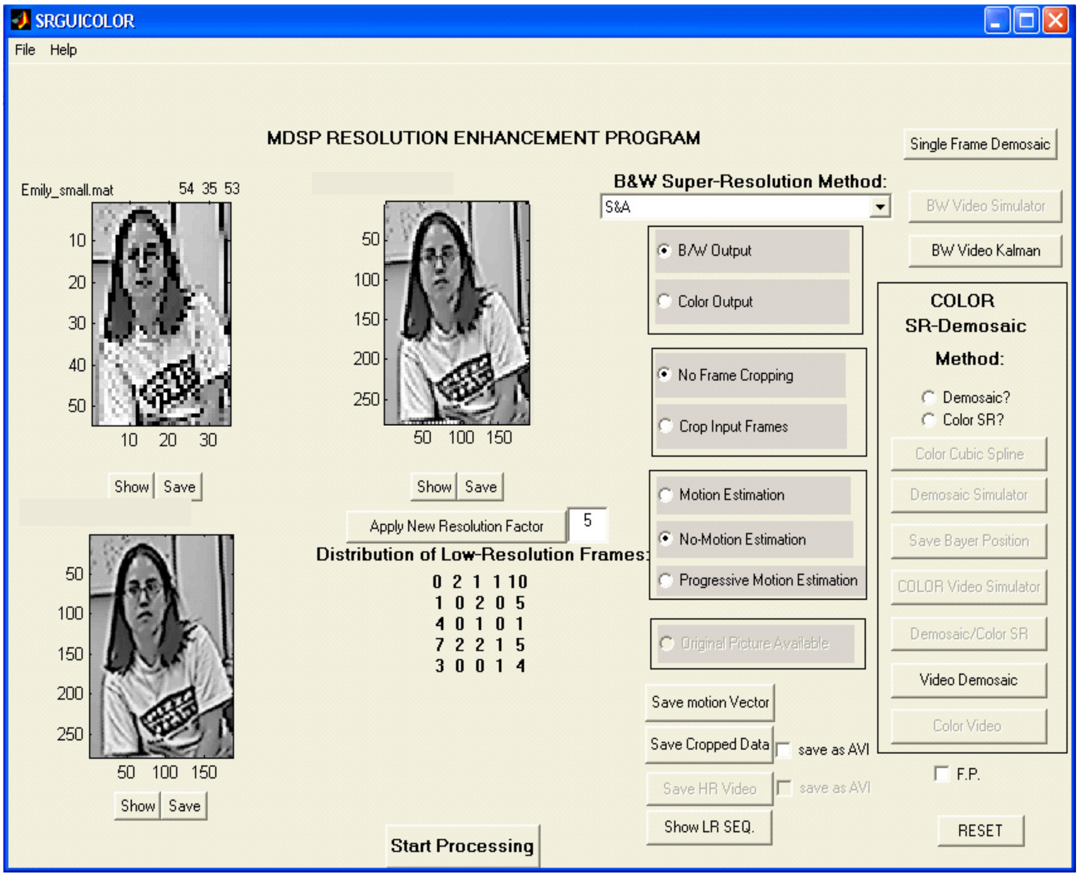
  - As part of this software package, motion estimation is done automatically by the program (pairwise anchoring, pairwise progressive, multi-frame projective, or multi-frame robust iterative) or independently estimated motion vectors may be provided by the user.

  - The user is able to specify the region of interest to be processed.
  - A basic tracking algorithm is incorporated in the program so that if only a certain part of the input images are important for the user (a car moving in a crowded street), this region can be tracked and another data sequence containing only that particular object is produced.
  - The parameters of the imaging system (such as the point-spread function) may be specified by the user.
  - The input image files (as well as the output files) may be given as .mat (Matlab data file) or .avi format.
  - Producing color or grayscale output images are optional, given color (or raw CFA) input frames.
  - For purposes of experimentation, the software package is capable of producing simulated video data for different imaging scenarios from a single high resolution input image, with user-controlled parameters.

This software is currently licensed and used in tens of academic and industrial institutes. Figure 6.1 illustrates a screenshot of the MDSP software. More information on this software tool is available at *http://www.ee.ucsc.edu/∼milanfar/SR-Software.htm* .

6.2 Future Work

- One important extension for our algorithms is the incorporation of blur identification algorithms in the super-resolution methods. Many single-frame blind deconvolution algorithms have been suggested in the last 30 years [92, 93]; and recently, [27, 94] incorporated a single parameter blur identification algorithm in their super-resolution method.



SRGUICOLOR

MDSP RESOLUTION ENHANCEMENT PROGRAM

Emily_small.mat 54 35 53

B&W Super-Resolution Method: S&A

● B/W Output
○ Color Output

● No Frame Cropping
○ Crop Input Frames

○ Motion Estimation
● No-Motion Estimation
○ Progressive Motion Estimation

○ Original Picture Available

COLOR SR-Demosaic Method:
○ Demosaic?
○ Color SR?

Apply New Resolution Factor 5

Distribution of Low-Resolution Frames:

```
0 2 1 1 10
1 0 2 0 5
4 0 1 0 1
7 2 2 1 5
3 0 0 1 4
```

□ save as AVI
□ save as AVI
□ F.P.

Figure 6.1: Screenshot of the software package that is based on the material presented in this thesis.

Still, there is need for more research to provide a super-resolution method along with a more general blur estimation algorithm.

- In Section 2.2.5, we proposed and mathematically justified a very fast two-step "Shift-and-Add" super-resolution method when relative motion is pure translational, and PSF is common and space-invariant in all low-resolution images. Empirical results show that this computationally efficient two-step method can be also applied to the cases with more complicated motion. The quality of reconstruction depends on the validity of the commuting property for the motion and blur matrices,

$$HF_k - F_kH \approx \mathbf{0} \quad . \tag{6.1}$$

There is need for more research on defining the conditions under which (6.1) is valid.

- One of the most apparent effects of DCT based compression methods, such as *MPEG* for video and *JPEG* for still images, is the blocking artifact. The quantization noise variance of each pixel in a block is space-varying. For a block located in a low-frequency content area, pixels near boundaries contain more quantization noise than the interior pixels. On the other hand, for the blocks located in the high-frequency area, pixels near boundaries contain less quantization noise than the interior pixels [95]. This space-variant noise property of the blocks may be exploited to reduce the quantization noise. Because of the presence of motion in video sequences, pixel locations in the blocks change from one frame to the other. So two corresponding pixels from two different frames may be located on and off the boundaries of the DCT blocks in which they are located. Based on the discussion that was presented in the previous chapters, it is relatively easy to determine which pixel has less quantization noise. It is reasonable to assign a higher weight to those pixels which suffer less from quantization noise in the data fusion step, as explained in Section 2.2.5. The relative magnitude of the weight assigned because of quantization and the weight that was explained in Section 2.2.5 will depend on the compression ratio.

- Few papers [34, 35] have addressed resolution enhancement of compressed video sequences. Compression artifacts can dramatically decrease the performance of any super-resolution system. Considering compression color artifacts in designing novel multi-frame demosaicing algorithms is part of our ongoing work.

- Study of different compression techniques and their effects on the quality of reconstruction is not only essential for the optimal reconstruction of super-resolved images from compressed data, but also it is very important for designing novel compression techniques. A very interesting extension to our research is to focus on the design of a novel compression method which results in compressed low-quality images ideal for reconstruction by its matching super-resolution technique. Such method is of great importance for the design of efficient HD-TV video streams.

- Accurate subpixel motion estimation is an essential part of any image fusion process such as multi-frame super-resolution or demosaicing. To the best of our knowledge, no paper has properly addressed the problem of estimating motion between Bayer filtered images (an ad-hoc registration method for color filtered images is suggested in [69]). However, a few papers have addressed related issues. A general method for aligning images from sensors of different modalities based on local-normalized-correlation technique is proposed in [96], however no experiment is offered to attest to its subpixel accuracy. Ref. [72] has addressed the problem of color motion estimation, where information from different color channels are incorporated by simply using alternative color representations such as HSV or normalized RGB. More work remains to be done to fully analyze subpixel motion estimation from color-filtered images. Moreover, a few papers have suggested different methods of concurrent motion estimation and [35, 50, 51, 97–99]. Simulation results show the effectiveness of these methods. Therefore an important extension of our research includes incorporation of motion estimation algorithms in the proposed multi-frame demosaicing method of Chapter 3.

- While the proposed static super-resolution and demosaicing methods are applicable to a very wide range of data and motion models, our dynamic SR method is developed for the case of translational motion and common space-invariant blur. A fast and robust recursive data fusion algorithm based on the L_1 norm minimization applicable to general motion models is a promising extension to this work.

- In the previous chapter we studied several multi-frame motion estimation techniques and presented a robust optimal framework for addressing this problem. We showed that in some cases using multi-frame motion estimation techniques are very useful and helps improve the results of the associating multi-frame image fusion methods. However, in general, any multi-frame image fusion technique is computationally much more complex than the corresponding single-frame method. Therefore, it is reasonable to use such methods only in cases for which one expects significant improvement in estimation accuracy.

The performance of the single-frame motion estimation methods are studied in many previous works, including [80, 100–104]. However, beside the very recent works of [105, 106] which study the fundamental performance limits (Cramér-Rao Bounds [41]) of the non-constrained multi-frame image registration problems, to the best of our knowledge there is not much work on the performance of the multi-frame constrained motion estimation method. As a part of future work, by thoroughly analyzing the the performance of the multi-frame motion estimation methods, we can define the cases for which such methods can be useful.

- Recently, the challenge of simultaneous resolution enhancement in time as well as space has received growing attention [107–109]. A problem worthy of further research is to apply features such as motion consistency, robustness, and computational efficiency to this unified space-time resolution enhancement model.

6.3 Closing

In this thesis, theory, analysis, and justification for a novel framework of fast and robust algorithms, applicable to a variety of multi-frame image fusion problems are developed. Experiments on both simulated and real data demonstrate the effectiveness of the presented method, and its practicality for real applications. The author hopes that the work presented here serves as a stepping stone for the future scholars in this exciting field of research.

Appendix A

The Bilateral Filter

The idea of the bilateral filter was first proposed in [110] as a very effective one-pass filter for denoising purposes, while keeping edges sharp. Unlike conventional filters, the bilateral filter defines the closeness of two pixels not only based on geometric distance but also based on photometric distance. Considering the 1-D case (to simplify the notations), the result of applying the bilateral filter for the k^{th} sample in the estimated (1-D) signal $\widehat{\underline{X}}$ is

$$[\widehat{\underline{X}}]_k = \frac{\sum_{m=-M}^{M} W[k,m][\underline{Y}]_{k-m}}{\sum_{m=-M}^{M} W[k,m]}, \tag{A.1}$$

where $\underline{Y} = \underline{X} + \underline{V}$ is the noisy image (vector), and $2M+1$ is the size of 1-D bilateral kernel. The weight $W[k,m] = W_S[k,m]W_P[k,m]$ considers both photometric and spatial difference between sample k in the noisy vector $\underline{Y}$ and its neighbors to define the value of sample k in the estimated vector $\widehat{\underline{X}}$. The spatial and photometric difference weights were arbitrarily defined in [110] as

$$W_S[k,m] = \exp\left\{-\frac{m^2}{2\sigma_S^2}\right\},$$

$$W_P[k,m] = \exp\left\{-\frac{([\underline{Y}]_k - [\underline{Y}]_{k-m})^2}{2\sigma_R^2}\right\}, \tag{A.2}$$

where parameters σ_S^2 and σ_R^2 control the strength of spatial and photometric property of the filter respectively.

In [111], Elad proved that such filter is a single Jacobi iteration of the following weighted least-squares minimization

$$\begin{aligned}\underline{\widehat{X}} &= \underset{\underline{X}}{\text{ArgMin}} \left[\|\underline{X} - \underline{Y}\|_2^2 + \lambda \sum_{m=-M}^{M} \|\underline{X} - S^m \underline{X}\|_{\mathbf{W_m}}^2 \right] \\ &= \underset{\underline{X}}{\text{ArgMin}} \left[[\underline{X} - \underline{Y}]^T [\underline{X} - \underline{Y}] + \lambda \sum_{m=-M}^{M} [\underline{X} - S^m \underline{X}]^T \, \mathbf{W_m} \, [\underline{X} - S^m \underline{X}] \right], \quad \text{(A.3)}\end{aligned}$$

where S^m implies a shift to the right of m samples, and $\mathbf{W_m}$ is defined from (A.2). He also showed that using more iterations will enhance the performance of this filter.

Note that if we define the $(i,i)^{\text{th}}$ element of the diagonal weight matrix $\mathbf{W_m}$ as

$$\mathbf{W_m}(i,i) = \frac{\alpha^m}{|\underline{X}(i) - S^m \underline{X}(i)|} \qquad 0 < \alpha < 1 \quad ,$$

that is, weighting the estimate with respect to both photometric distance $|\underline{X}(i) - S^n \underline{X}(i)|$ and geometric distance α^m, then (A.3) will become

$$\underline{\widehat{X}} = \underset{\underline{X}}{\text{ArgMin}} \left[\|\underline{X} - \underline{Y}\|_2^2 + \lambda \sum_{m=-M}^{M} \alpha^m \|\underline{X} - S^m \underline{X}\|_1 \right],$$

which is the 1-D version of the B-TV criterion in (2.16).

Appendix B

The Limitations and Improvement of the Zomet Method [1]

A robust super-resolution method was recently proposed by Zomet et al. in [1], where robustness is achieved by modifying the gradient of the L_2 norm cost function (2.7):

$$\begin{aligned}\underline{G}_2 =& \sum_{k=1}^{N} \underline{B}(k) = \sum_{k=1}^{N} F^T(k) H^T(k) D^T(k) \left(D(k)H(k)F(k)\underline{X} - \underline{Y}(k) \right) \\ =& \sum_{k=1}^{N} F^T(k) H^T(k) D^T(k) \underline{U}(k),\end{aligned} \tag{B.1}$$

in which $\underline{B}(k)$ is the gradient resulted from frame k and $\underline{U}(k)$ represents the residual vector. They substituted (B.1) with the following

$$\widehat{\underline{G}}_2 = \text{MED}\{\underline{B}(k)\}_{k=1}^{N} = \text{MED}\{F^T(k) H^T(k) D^T(k) \underline{U}(k)\}_{k=1}^{N}, \tag{B.2}$$

where MED is a pixelwise median operator (instead of the mean operator in (B.1)). Then steepest descent minimization was used to calculate $\widehat{\underline{X}}$

$$\widehat{\underline{X}}_{n+1} = \widehat{\underline{X}}_n + \lambda'' \widehat{\underline{G}}_2, \tag{B.3}$$

where λ'' is the step size in the direction of the gradient.

We show that for certain imaging scenarios the approximated gradient (B.2) is zero in all iterations, which means estimated high-resolution frame of the n^{th} iteration (X_n) is the same as the initial guess (X_0) and the method “stalls” and fails. To appreciate this fact, let us start with a square case in which the blurring effect is negligible (i.e. H_k is an identity matrix resulting in $\underline{B}(k) = F^T(k)D^T(k)\underline{U}(k)$). A quick consultation with Figure 2.3 suggests that only one of every r^2 elements in $D^T(k)\underline{U}(k)$ has a non-zero value. Moreover, recall that $F^T(k)$ just registers the vector $D^T(k)\underline{U}(k)$ with respect to the estimated relative motion without changing its value. According to (B.2), $\underline{\widehat{g}}(i)$ (the i^{th} element of the gradient vector $\underline{\widehat{G}_2}$) is equal to[1] MED$\{\underline{b}_i(k)\}_{k=1}^{N}$. As $N-1$ elements in $\{\underline{b}_i(k)\}_{k=1}^{N}$ have zero value, their median will also be zero. Therefore every element of the approximated gradient vector will be zero. Even for a more general case in which the effect of blur is not negligible ($H(k)$ is a matrix form of the $m \times n$ blur kernel), the same approach may be employed to show that unless $(m \times n > \frac{r^2}{2})$, the gradient will remain zero for all iterations.

The $(m \times n > \frac{r^2}{2})$ condition is also valid for the over-determined cases where the distribution of motion vectors is uniform (that is the number of available low-resolution measurements for each pixel in the high-resolution grid is equal). Therefore this condition does not depend on the number of available low-resolution frames. In particular, consider the identity blur matrix case, where the addition of any new frame $Y(\vartheta)$ is equivalent to the addition of a new gradient vector $\underline{B}(\vartheta)$ with r^2-1 times more zero elements (resulting from upsampling) than non-zero elements to the stack of gradient vectors. Therefore if

$$\underline{\widehat{g}}(i) = \text{MED}\{\underline{b}_i(k)\}_{k=1}^{N} = \underline{0},$$

even after addition of r^2 uniformly spread low-resolution frames $\underline{\widehat{g}}(i) = \text{MED}\{\underline{b}_i(k)\}_{k=1}^{N+r^2}$ will still be zero (as r^2-1 values of r^2 newly added elements are zeros). Generalization of this property to the case of arbitrary number of low-resolution frames with uniform motion distribution is straight forward.

This limitation can be overcome by modifying the MED operator in (B.2). This modified median operator would not consider those elements of $\underline{b}_i(k)$ which are the result of zero fill-

[1] $\underline{b}_i(k)$ is the i^{th} element of the vector $\underline{B}(k)$.

ing. It is interesting to note that such an assumption will result in estimating the high-resolution frame as the median of registered low-resolution frames after zero filling, which is the exact interpretation of using L_1 norm minimization discussed in Section 2.2.2. In essence, we can say that this “corrected” version of the algorithm in [1] is a special case of our more general robust formulation.

Appendix C

Noise Modeling Based on GLRT Test

In this appendix we explain our approach of deciding which statistical model better describes the probability density function (PDF) of the noise. Gaussian and Laplacian distributions, the two major candidates for modeling the noise PDF, are defined as

$$P_G(\underline{V}) = \frac{1}{(2\pi\sigma_G^2)^{Q/2}} \exp(-\frac{\sum_{i=1}^{Q}([\underline{V}]_i - m_G)^2}{2\sigma_G^2}), \tag{C.1}$$

$$P_L(\underline{V}) = \frac{1}{(2\sigma_L)^Q} \exp(-\frac{\sum_{i=1}^{Q}|[\underline{V}]_i - m_L|}{\sigma_L}), \tag{C.2}$$

where $[\underline{V}]_i$ is the i^{th} element of the noise vector $\underline{V}$ (of size $[1 \times Q]$) and σ_G, m_G are the unknown parameters of the Gaussian PDF (P_G) and σ_L, m_L are the unknown parameters of the of the Laplacian PDF (P_L) which are estimated from data. Noting that logarithm is a monotonic function and

$$\ln P_L(\underline{V}) = -Q \ln 2 - Q \ln \sigma_L - \frac{\sum_{i=1}^{Q}|[\underline{V}]_i - m_L|}{\sigma_L}, \tag{C.3}$$

then the ML estimates of σ_L and m_L are calculated as

$$\widehat{\sigma}_L, \widehat{m}_L = \underset{\sigma_L, m_L}{\text{ArgMax}}(P_L(\underline{V})) = \underset{\sigma_L, m_L}{\text{ArgMax}}(-\ln P_L(\underline{V})), \tag{C.4}$$

so

$$-\frac{\partial \ln P_L(\underline{V})}{\partial m_L} = \sum_{i=1}^{Q} |[\underline{V}]_i - m_L| = 0 \Longrightarrow \widehat{m}_L = \text{MEDIAN}(\underline{V}), \tag{C.5}$$

and

$$-\frac{\partial \ln P_L(\underline{V})}{\partial \sigma_L} = -\frac{Q}{\sigma_L} + \frac{\sum_{i=1}^{Q} |[\underline{V}]_i - m_L|}{\sigma_L^2} = 0 \Longrightarrow \widehat{\sigma}_L = \frac{\sum_{i=1}^{Q} |[\underline{V}]_i - \widehat{m}_L|}{Q}. \qquad \text{(C.6)}$$

The same scheme can be used to estimate the Gaussian model parameters as:

$$\widehat{m}_G = \text{MEAN}(\underline{V}) \quad \text{and} \quad \widehat{\delta}_G = \sqrt{\frac{\sum_{i=1}^{Q}([\underline{V}]_i - \widehat{m}_G)^2}{Q}} \ . \qquad \text{(C.7)}$$

We use the generalized likelihood ratio test (GLRT) [112] to decide between the two hypotheses about the noise model:

$$\frac{P_G(\underline{V}; \widehat{\sigma}_G, \widehat{m}_G)}{P_L(\underline{V}; \widehat{\sigma}_L, \widehat{m}_L)} > \gamma, \qquad \text{(C.8)}$$

where γ is the decision threshold. That is if the ratio in (C.8) is larger than γ then P_G is a more accurate PDF model for $\underline{V}$ than P_L and vice versa (γ was chosen equal to 1 as it mimics a test which minimizes the probability of error and does not a priori favor either hypothesis). So:

$$\frac{\frac{1}{(2\pi\widehat{\sigma}_G^2)^{Q/2}} \exp(-\frac{\sum_{i=1}^{Q}([\underline{V}]_i - \widehat{m}_G)^2}{2\widehat{\sigma}_G^2})}{\frac{1}{(2\widehat{\sigma}_L)^Q} \exp(-\frac{\sum_{i=1}^{Q}|[\underline{V}]_i - \widehat{m}_L|}{\widehat{\sigma}_L})} > 1 \ . \qquad \text{(C.9)}$$

Substituting $\widehat{m}_G$, $\widehat{\sigma}_L$, $\widehat{\sigma}_G$, and $\widehat{\sigma}_L$ with their corresponding estimates from (C.5), (C.6), and (C.7) and simplifying results in

$$\frac{\widehat{\sigma}_L}{\widehat{\sigma}_G} > (\frac{\pi}{2e})^{\frac{1}{2}} \simeq 0.7602 \ . \qquad \text{(C.10)}$$

So if (C.10) is valid for a certain vector $\underline{V}$ then the Gaussian is a better model of data than the Laplacian model, and vice versa.

Appendix D

Error Modeling Experiment

The Maximum Likelihood estimators of the high-resolution image developed in many previous works [11, 113] are valid when the noise distribution follows the Gaussian model. Unfortunately, the Gaussian noise assumption is not valid for many real world image sequences. To appreciate this claim we set up the following experiments. In these experiments according to the model in (2.4) a high-resolution $[256 \times 256]$ image was shifted, blurred, and downsampled to create 16 low-resolution images (of size $[64 \times 64]$). The effect of readout noise of CCD pixels was simulated by adding Gaussian noise to these low-resolution frames achieving SNR equal to 25dB. We considered three common sources of error (outliers) in super-resolution reconstruction:

1. Error in motion estimation.
2. Inconsistent pixels: effect of an object which is only present in a few low-resolution frames (e.g. the effects of a flying bird in a static scene).
3. Salt and Pepper noise.

In the first experiment, to simulate the effect of error in motion estimation, a bias equal to $\frac{1}{4}$ of a pixel was intentionally added to the known motion vector of only one of the low-resolution frames. In the second experiment, a $[10 \times 10]$ block of only one of the images was replaced by a block from another data sequence. And finally, in the third experiment, we added salt and pepper noise to approximately 1% of the pixels of only one of the low-resolution frames. We used the GLRT test (Appendix C) to compare the goodness of fit of Laplacian and Gaussian distributions for modeling the noise in these three sets of low-resolution images. Consider the general model of (2.4), the overall noise (error residual) is defined as

$$\underline{V}=\begin{bmatrix}\underline{V}(1)\\ \underline{V}(2)\\ \vdots\\ \underline{V}(N)\end{bmatrix},$$

$$\underline{V}(k)=\underline{Y}(k)-D(k)H(k)F(k)\underline{X}, \tag{D.1}$$

where N is the number of the frames in the sequence.

The GLRT test results for these three experiments were 0.6084, 0.6272 and 0.6081, respectively. The test result for the original low-resolution images contaminated only with pure Gaussian noise was 0.7991. Based on the criterion in (C.10), the distribution of the noise with a test result smaller than 0.7602 is better modeled by the Laplacian model rather than the Gaussian model. Note that the outlier contamination in these tests was fairly small, and more outlier contamination (larger error in motion estimation, larger blocks of inconsistence pixels, and higher percentage of Salt and Pepper noise) results in even smaller GLRT test results.

Appendix E

Derivation of the Inter-Color Dependencies Penalty Term

In this appendix, we illustrate the differentiation of the first term in (3.10), which we call $\underline{L}$, with respect to $\underline{X}_G$. From (3.10) we have:

$$\underline{L} = \|\underline{X}_G \odot S_x^l S_y^m \underline{X}_B - \underline{X}_B \odot S_x^l S_y^m \underline{X}_G\|_2^2 \xrightarrow[\text{commutative}]{\odot \text{ is}} \underline{L} = \|S_x^l S_y^m \underline{X}_B \odot \underline{X}_G - \underline{X}_B \odot S_x^l S_y^m \underline{X}_G\|_2^2.$$

We can substitute the element by element multiplication operator "$\odot$", with the differentiable dot product by rearranging $\underline{X}_B$ as the diagonal matrix[1] $\mathbf{X_B}$ and $S_x^l S_y^m \underline{X}_B$ as $\mathbf{X_B^{l,m}}$, which is the diagonal form of the shifted $\underline{X}_B$ by l, m pixels in horizontal and vertical directions,

$$\underline{L} = \|\mathbf{X_B^{l,m}} \underline{X}_G - \mathbf{X_B} S_x^l S_y^m \underline{X}_G\|_2^2. \tag{E.1}$$

Using the identity:

$$\frac{\partial \|\mathbf{\Delta}\underline{C}\|_2^2}{\partial \underline{C}} = \frac{\partial \left(\underline{C}^T \mathbf{\Delta^T \Delta} \underline{C}\right)}{\partial \underline{C}} = 2\mathbf{\Delta^T \Delta}\underline{C},$$

[1] We are simply mapping a vector $\underline{\Delta}$ to its diagonal matrix representation $\mathbf{\Delta}$ such that:

$$\underline{\Delta} = \begin{pmatrix} \Delta_1 \\ \Delta_2 \\ \vdots \\ \Delta_{4r^2Q_1Q_2} \end{pmatrix} \longrightarrow \begin{pmatrix} \Delta_1 & 0 & \cdots & 0 \\ 0 & \Delta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Delta_{4r^2Q_1Q_2} \end{pmatrix} = \mathbf{\Delta}$$

and noting that $\mathbf{X_B^{l,m}}$ and $\mathbf{X_B}$ are symmetric matrices, the differentiation with respect to the green band will be computed as follows

$$\frac{\partial \underline{L}}{\partial \underline{X}_G} = 2(\mathbf{X_B^{l,m}} - S_x^{-l} S_y^{-m} \mathbf{X_B})(\mathbf{X_B^{l,m}} \underline{X}_G - \mathbf{X_B} S_x^l S_y^m \underline{X}_G).$$

Differentiation of the second term in (3.10), and also differentiation with respect to the other color bands follow the same technique.

Appendix F

Appendix: Affine Motion Constraints

In this section we review the consistency constraints for the affine motion model. The affine transformation models a composition of rotation, translation, scaling, and shearing. This six parameter global motion model is defined by

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} a_{i,j} & b_{i,j} \\ c_{i,j} & d_{i,j} \end{bmatrix} \begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} e_{i,j} \\ f_{i,j} \end{bmatrix}, \tag{F.1}$$

where $[x_i, y_i]^T$, and $[x_j, y_j]^T$ are the coordinates of two corresponding pixels in frames i and j. Defining

$$\Xi_{i,j} = \begin{bmatrix} a_{i,j} & b_{i,j} \\ c_{i,j} & d_{i,j} \end{bmatrix}, \quad \underline{T}_{i,j} = \begin{bmatrix} e_{i,j} \\ f_{i,j} \end{bmatrix}, \tag{F.2}$$

the consistency constraints for the affine case are defined by the relations

$$\forall \quad 1 \leq i, j, k \leq N, \quad \begin{cases} \Xi_{i,k} = & \Xi_{i,j}\Xi_{j,k} \\ \underline{T}_{i,k} = & \Xi_{i,j}\underline{T}_{j,k} + \underline{T}_{i,j} \end{cases}. \tag{F.3}$$

Note that $\Xi_{i,i} = I$ and $\underline{T}_{i,i} = \underline{0}$, and therefore (F.3) results in a set of $6(N-1)^2$ independent nonlinear constraints.

A more intuitive (and perhaps more practical) set of constrains can be obtained if we consider a simplified version of the general affine model where only scale, rotation, and translation are considered. Such a simplified model is represented by replacing the first coefficient matrix on the right side of (F.1) with

$$\Xi'_{i,j} = \begin{bmatrix} a_{i,j} & b_{i,j} \\ c_{i,j} & d_{i,j} \end{bmatrix} = \kappa_{i,j} \begin{bmatrix} \cos(\theta_{i,j}) & -\sin(\theta_{i,j}) \\ \sin(\theta_{i,j}) & \cos(\theta_{i,j}) \end{bmatrix}, \tag{F.4}$$

where $\kappa_{i,j}$, and $\theta_{i,j}$ are the scaling and rotation parameters, respectively. The consistency constraints for this simplified affine model are given by the following relations

$$\begin{cases} \kappa_{i,k} = & \kappa_{i,j}\kappa_{j,k} \\ \theta_{i,k} = & \theta_{i,j} + \theta_{j,k} \\ \underline{T}_{i,k} = & \Xi'_{i,j}\underline{T}_{j,k} + \underline{T}_{i,j} \end{cases} . \tag{F.5}$$

For a set of N frames, the above relations amount to $4(N-1)^2$ independent non-linear constraints. Non-linear programming (e.g. “*fmincon*” function in MATLAB) can be used to minimize the cost functions with such non-linear constraints.

Bibliography

[1] A. Zomet, A. Rav-Acha, and S. Peleg, "Robust super resolution," in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 645–650, Dec. 2001.

[2] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Processing*, vol. 8, pp. 1221–1228, Sept. 1999.

[3] C. Laroche and M. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients." United States Patent 5,373,322, 1994.

[4] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multi-frame super-resolution," *IEEE Trans. Image Processing*, vol. 13, pp. 1327–1344, Oct. 2004.

[5] K. A. Parulski, L. J. D'Luna, B. L. Benamati, and P. R. Shelley, "High performance digital color video camera," *J. Electron. Imaging*, vol. 1, pp. 35–45, Jan. 1992.

[6] H. H. Barrett and K. J. Myers, *Foundations of Image Science.* Hoboken, New Jersey: John Wiley & Sons, 2003.

[7] S. Borman, *Topics in Multiframe Superresolution Restoration*. PhD thesis, University of Notre Dame, Notre Dame, IN, May 2004.

[8] T. S. Huang and R. Y. Tsai, "Multi-frame image restoration and registration," *Advances in computer vision and Image Processing*, vol. 1, pp. 317–339, 1984.

[9] C. R. Vogel, *Computational methods for inverse problems*. Frontiers in Applied Mathematics, Philadelphia, PA: SIAM, 2002.

[10] A. Patti, M. Sezan, and A. M. Tekalp;, "Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time," *IEEE Trans. Image Processing*, vol. 6, pp. 1326–1333, Mar. 1997.

[11] M. Elad and A. Feuer, "Restoration of single super-resolution image from several blurred, noisy and down-sampled measured images," *IEEE Trans. Image Processing*, vol. 6, pp. 1646–1658, Dec. 1997.

[12] G. Golub and C. V. Loan, *Matrix computations*. London: The Johns Hopkins University Press, third ed., 1996.

[13] M. A. Lukas, "Asymptotic optimality of generalized cross-validation for choosing the regularization parameter," *Numerische Mathematik*, vol. 66, no. 1, pp. 41–66, 1993.

[14] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Trans. Image Processing*, vol. 10, pp. 1299–1308, Sept. 2001.

[15] P. C. Hansen and D. P. O'Leary, "The use of the L-curve in the regularization of ill-posed problems," *SIAM J. Sci. Comput.*, vol. 14, pp. 1487–1503, Nov. 1993.

[16] A. Bovik, *Handbook of Image and Video Processing*. New Jersey: Academic Press Limited, 2000.

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

[18] R. R. Schultz and R. L. Stevenson, "Extraction of high-resolution frames from video sequences," *IEEE Trans. Image Processing*, vol. 5, pp. 996–1011, June 1996.

[19] N. K. Bose, H. C. Kim, and H. M. Valenzuela, "Recursive implementation of total least squares algorithm for image reconstruction from noisy, undersampled multiframes," in *Proc. of the IEEE Int. Conf. Acoustics, Speech, and Signal Processing(ICASSP)*, vol. 5, pp. 269–272, Apr. 1993.

[20] S. Borman and R. L. Stevenson, "Super-resolution from image sequences - a review," in *Proc. of the 1998 Midwest Symposium on Circuits and Systems*, vol. 5, Apr. 1998.

[21] L. Teodosio and W. Bender, "Salient video stills: Content and context preserved," in *Proc. of the First ACM Int. Conf. on Multimedia*, vol. 10, pp. 39–46, Aug. 1993.

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

[23] M. C. Chiang and T. E. Boulte, "Efficient super-resolution via image warping," *Image and Vision Computing*, vol. 18, pp. 761–771, July 2000.

[24] S. Peleg, D. Keren, and L. Schweitzer, "Improving image resolution using subpixel motion," *CVGIP:Graph. Models Image Processing*, vol. 54, pp. 181–186, March 1992.

[25] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP:Graph. Models Image Process*, vol. 53, pp. 231–239, 1991.

[26] H. Ur and D. Gross, "Improved resolution from sub-pixel shifted pictures," *CVGIP:Graph. Models Image Processing*, vol. 54, Mar. 1992.

[27] M. Ng and N. Bose, "Mathematical analysis of super-resolution methodology," *IEEE Signal Processing Mag.*, vol. 20, pp. 62–74, May 2003.

[28] S. Lertrattanapanich and N. K. Bose, "High resolution image formation from low resolution frames using Delaunay triangulation," *IEEE Trans. Image Processing*, vol. 11, pp. 1427–1441, Dec. 2002.

[29] S. C. Zhu and D. Mumford, "Prior learning and gibbs reaction-diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 1236–1250, Nov. 1997.

[30] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Tree-based resolution synthesis," in *Proc. of the IS&T Conf. on Image Processing, Image Quality, Image Capture Systems*, pp. 405–410, 1999.

[31] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1167–1183, Sept. 2002.

[32] E. Haber and L. Tenorio, "Learning regularization functionals-a supervised training approach," *Inverse Problems*, vol. 19, pp. 611–626, June 2003.

[33] M. Ben-Ezra, A. Zomet, and S. Nayar, "Video super-resolution using controlled subpixel detector shifts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 977–987, June 2004.

[34] Y. Altunbasak, A. Patti, and R. Mersereau, "Super-resolution still and video reconstruction from MPEG-coded video," *IEEE Trans. Circuits And Syst. Video Technol.*, vol. 12, pp. 217–226, Apr. 2002.

[35] C. A. Segall, R. Molina, A. Katsaggelos, and J. Mateos, "Bayesian high-resolution reconstruction of low-resolution compressed video," in *Proc. of IEEE Int. Conf. on Image Processing*, vol. 2, pp. 25–28, Oct. 2001.

[36] C. Segall, A. Katsaggelos, R. Molina, and J. Mateos, "Bayesian resolution enhancement of compressed video," *IEEE Trans. Image Processing*, vol. 13, pp. 898–911, July 2004.

[37] M. G. Kang and S. Chaudhuri, "Super-resolution image reconstruction," *IEEE Signal Processing Magazine*, vol. 20, pp. 21–36, May 2003.

[38] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, pp. 47–57, Aug. 2004.

[39] G. C. Calafiore, "Outliers robustness in multivariate orthogonal regression," *IEEE Trans. Syst. Man and Cybernetics*, vol. 30, pp. 674–679, Nov. 2000.

[40] P. J. Huber, *Robust Statitics*. New York: Wiley, 1981.

[41] S. M. Kay, *Fundamentals of statistical signal processing:estimation theory*, vol. I. Prentice-Hall, 1993.

[42] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Robust shift and add approach to super-resolution,” *Proc. of the 2003 SPIE Conf. on Applications of Digital Signal and Image Processing*, pp. 121–130, Aug. 2003.

[43] A. M. Tekalp, *Digital Video Processing*. Prentice-Hall, 1995.

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

[45] T. F. Chan, S. Osher, and J. Shen, “The digital TV filter and nonlinear denoising,” *IEEE Trans. Image Processing*, vol. 10, pp. 231–241, Feb. 2001.

[46] Y. Li and F. Santosa, “A computational algorithm for minimizing total variation in image restoration,” *IEEE Trans. Image Processing*, vol. 5, pp. 987–995, June 1996.

[47] A. Zomet and S. Peleg, “Efficient super-resolution and applications to mosaics,” in *Proc. of the Int. Conf. on Pattern Recognition (ICPR)*, pp. 579–583, Sept. 2000.

[48] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, “Hierachical model-based motion estimation,” *Proc. of the European Conf. on Computer Vision*, pp. 237–252, May 1992.

[49] S. Park, M. Park, and M. G. Kang, “Super-resolution image reconstruction, a technical overview,” *IEEE Signal Processing Magazine*, vol. 20, pp. 21–36, May 2003.

[50] N. R. Shah and A. Zakhor, “Resolution enhancement of color video sequences,” *IEEE Trans. Image Processing*, vol. 8, pp. 879–885, June 1999.

[51] B. C. Tom and A. Katsaggelos, “Resolution enhancement of monochrome and color video using motion compensation,” *IEEE Trans. Image Processing*, vol. 10, pp. 278–287, Feb. 2001.

[52] D. R. Cok, “Signal processing method and apparatus for sampled image signals.” United States Patent 4,630,307, 1987.

[53] J. Hamilton and J. Adams, “Adaptive color plan interpolation in single sensor color electronic camera.” United States Patent 5,629,734, 1997.

[54] L. Chang and Y.-P. Tan, “Color filter array demosaicking: new method and performance measures,” *IEEE Trans. Image Processing*, vol. 12, pp. 1194–1210, Oct. 2002.

[55] K. Hirakawa and T. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” in *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 3, pp. 669–672, Sept. 2003.

[56] D. Keren and M. Osadchy, “Restoring subsampled color images,” *Machine Vision and applications*, vol. 11, no. 4, pp. 197–202, 1999.

[57] Y. Hel-Or and D. Keren, “Demosaicing of color images using steerable wavelets,” Tech. Rep. HPL-2002-206R1 20020830, HP Labs Israel, 2002.

[58] W. K. Pratt, *Digital image processing*. New York: John Wiley & Sons, INC., 3rd ed., 2001.

[59] D. Taubman, “Generalized Wiener reconstruction of images from colour sensor data using a scale invariant prior,” in *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 3, pp. 801–804, Sept. 2000.

[60] D. D. Muresan and T. W. Parks, “Optimal recovery demosaicing,” in *IASTED Signal and Image Processing*, Aug. 2002.

[61] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections,” *IEEE Trans. Image Processing*, vol. 11, pp. 997–1013, Sep. 2002.

[62] S. C. Pei and I. K. Tam, “Effective color interpolation in CCD color filter arrays using signal correlation,” *IEEE Trans. Image Processing*, vol. 13, pp. 503–513, June 2003.

[63] D. Alleysson, S. Süsstrunk, and J. Hérault, “Color demosaicing by estimating luminance and opponent chromatic signals in the Fourier domain,” in *Proc. of the IS&T/SID 10th Color Imaging Conf.*, pp. 331–336, Nov. 2002.

[64] X. Wu and N. Zhang, “Primary-consistent soft-decision color demosaic for digital cameras,” in *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 1, pp. 477–480, Sept. 2003.

[65] R. Ramanath and W. Snyder, “Adaptive demosaicking,” *Journal of Electronic Imaging*, vol. 12, pp. 633–642, Oct. 2003.

[66] S. Farsiu, M. Elad, and P. Milanfar, “Multi-frame demosaicing and super-resolution from under-sampled color images,” *Proc. of the 2004 IS&T/SPIE Symp. on Electronic Imaging*, vol. 5299, pp. 222–233, Jan. 2004.

[67] D. Keren and A. Gotlib, “Denoising color images using regularization and correlation terms,” *Journal of Visual Communication and Image Representation*, vol. 9, pp. 352–365, Dec. 1998.

[68] A. Zomet and S. Peleg, “Multi-sensor super resolution,” in *Proc. of the IEEE Workshop on Applications of Computer Vision*, pp. 27–31, Dec. 2001.

[69] T. Gotoh and M. Okutomi, "Direct super-resolution and registration using raw CFA images," in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 600–607, July 2004.

[70] R. Ramanath, W. Snyder, G. Bilbro, and W. Sander, "Demosaicking methods for the Bayer color arrays," *Journal of Electronic Imaging*, vol. 11, pp. 306–315, July 2002.

[71] X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, "Color image quality metric S-CIELAB and its application on halftone texture visibility," in *IEEE COMPCON97 Symposium Digest.*, pp. 44–48, May 1997.

[72] P. Golland and A. M. Bruckstein, "Motion from color," *Computer Vision and Image Understanding*, vol. 68, pp. 346–362, Dec. 1997.

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

[74] M. Elad and A. Feuer, "Super-resolution reconstruction of image sequences," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp. 817–834, Sept. 1999.

[75] S. Farsiu, M. Elad, and P. Milanfar, "Multi-frame demosaicing and super-resolution of color images," *To appear in IEEE Trans. Image Processing*, vol. 15, Jan. 2005.

[76] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York: Academic Press, 1970.

[77] M. Elad, *Super resolution Reconstruction of Continuous Image Sequence*. PhD thesis, The Technion - Israel Institute of Technology, Haifa - Israel, 1997.

[78] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of dynamic linear systems," *AIAA Journal*, vol. 3, pp. 1445–1450, Aug. 1965.

[79] A. C. Harvey, *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge Univ Pr, 1990.

[80] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, pp. 325–376, December 1992.

[81] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. of DARPA Image Understanding Workshop*, pp. 121–130, 1981.

[82] W. Zhao and H. Sawhney, "Is super-resolution with optical flow feasible?," in *ECCV*, vol. 1, pp. 599–613, 2002.

[83] M. Alkhanhal, D. Turaga, and T. Chen, "Correlation based search algorithms for motion estimation," in *Picture Coding Symposium*, Apr. 1999.

[84] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Dynamic demosaicing and color superresolution of video sequences,” in *Proc. SPIE’s Conf. on Image Reconstruction from Incomplete Data III*, pp. 169–178, Denver, CO. Aug. 2004.

[85] S. Marcus and A. Willsky, “Algebraic structure and finite dimensional nonlinear estimation,” *SIAM J. Math. Anal.*, pp. 312–327, Apr. 1978.

[86] V. Govindu, “Lie-algebraic averaging for globally consistent motion estimation,” in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 684–691, July 2004.

[87] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, “Bundle adjustment – a modern synthesis,” in *Vision Algorithms: Theory and Practice*, vol. 1883 of *Lecture Notes in Computer Science*, pp. 298–372, 2000.

[88] H. Sawhney, S. Hsu, and R. Kumar, “Robust video mosaicing through topology inference and local to global alignment,” in *ECCV*, vol. 2, pp. 103–119, 1998.

[89] V. Govindu, “Combining two-view constraints for motion estimation,” in *Proc. of the Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 218–225, July 2001.

[90] Y. Sheikh, Y. Zhai, and M. Shah, “An accumulative framework for the alignment of an image sequence,” in *ACCV*, Jan. 2004.

[91] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation, and Time Series Analysis*. New York: Addison-Wesley Publishing Co., 1991.

[92] S. M. Jefferies and J. C. Christou, “Restoration of astronomical images by iterative blind deconvolution,” *Astrophysical Journal*, vol. 415, pp. 862–874, October 1993.

[93] D. Kondur and D. Hatzinakos, “Blind image deconvolution,” *IEEE Signal Processing Mag.*, vol. 13, pp. 43–64, May 1996.

[94] M. E. Tipping and C. M. Bishop, “Bayesian image super-resolution,” *Advances in Neural Information Processing Systems*, vol. 15, pp. 1303–1310, 2002.

[95] M. Robertson and R. Stevenson, “DCT quantization noise in compressed images,” in *Proc. of the IEEE Int. Conf. on Image Processing*, vol. 1, pp. 185–1888, Oct. 2001.

[96] M. Irani and P. Anandan, “Robust multi-sensor image alignment,” in *Proc. of IEEE Int. Conf. on Computer Vision*, pp. 959–966, Jan. 1998.

[97] R. Hardie, K. J. Barnard, and E. E. Armstrong, “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images,” *IEEE Trans. Image Processing*, vol. 6, pp. 1621–1633, Dec. 1997.

[98] R. R. Schultz, L. Meng, and R. Stevenson, "Subpixel motion estimation for super-resolution image sequence enhancement," *Journal of Visal Communication and Image Representation*, vol. 9, pp. 38–50, Mar. 1998.

[99] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "Double resolution from a set of aliased images," in *Proc. IS&T/SPIE Electronic Imaging 2004: Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications V*, vol. 5301, pp. 374–382, 2004.

[100] J. Barron, D. Fleet, S. Beauchemin, and T. Burkitt, "Performance of optical flow techniques," *CVPR*, vol. 92, pp. 236–242, 1992.

[101] W. F. Walker and G. E. Trahey, "A fundamental limit on the performance of correlation based phase correction and flow estimation techniques," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 41, pp. 644–654, September 1994.

[102] S. Auerbach and L. Hauser, "Cramér-Rao bound on the image registration accuracy," *Proceedings of SPIE*, vol. 3163, pp. 117–127, July 1997.

[103] H. Faroosh, J. Zerubia, and M. Berthod, "Extension of phase correlation to subpixel registration," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 188–200, 2002.

[104] D. Robinson and P. Milanfar, "Fundamental performance limits in image registration," *IEEE Transactions on Image Processing*, vol. 13, pp. 1185–1199, September 2004.

[105] D. Robinson and P. Milanfar, "Statistical performance analysis of super-resolution," *To Appear in IEEE Transactions on Image Processing*, 2005.

[106] T. Pham, M. Bezuijen, L. van Vliet, K. Schutte, and C. L. Hendriks, "Performance of optimal registration estimators," *Proceedings of SPIE Defense and Security Symposium*, vol. 5817, pp. 133–144, May 2005.

[107] M. A. Robertson and R. L. Stevenson, "Temporal resolution enhancement in compressed video sequences," *EURASIP Journal on Applied Signal Processing*, pp. 230–238, Dec. 2001.

[108] E. Shechtman, Y. Caspi, and M. Irani, "Increasing space-time resolution in video," in *Proc. of the European Conf. on Computer Vision (ECCV)*, pp. 331–336, May 2002.

[109] E. S. Y. Wexler and M. Irani, "Space-time video completion," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 120–127, June 2004.

[110] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. of IEEE Int. Conf. on Computer Vision*, pp. 836–846, Jan. 1998.

[111] M. Elad, "On the bilateral filter and ways to improve it," *IEEE Trans. Image Processing*, vol. 11, pp. 1141–1151, Oct. 2002.

[112] S. M. Kay, *Fundamentals of statistical signal processing:detection theory*, vol. II. Englewood Cliffs, New Jersey: Prentice-Hall, 1998.

[113] D. Capel and A. Zisserman, “Super-resolution enhancement of text image sequences,” in *Proc. of the Int. Conf. on Pattern Recognition*, pp. 1600–1605, 2000.

Consummatum Est.