NUMERICAL ALGORITHMS
FOR IMAGE SUPERRESOLUTION

A DISSERTATION
SUBMITTED TO THE PROGRAM IN
SCIENTIFIC COMPUTING AND COMPUTATIONAL MATHEMATICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Nhat Xuan Nguyen
July 2000

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Gene Golub
(Principal Co-Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Peyman Milanfar
(Principal Co-Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Michael Saunders

Approved for the University Committee on Graduate Studies:

Abstract

Image superresolution refers to image processing algorithms which produce high quality, high-resolution (HR) images from a set of low quality, low-resolution (LR) images. In many visual applications, both civilian and military, the imaging sensors have poor resolution outputs. When resolution can not be improved by replacing sensors, either because of cost or hardware physical limits, we can resort to superresolution algorithms. Even when superior equipment is available, superresolution algorithms are an inexpensive alternative.

Superresolution is a computationally intensive process. Some video applications may require superresolution to be done on-the-fly; data must be processed as they are received. To that end, it must take advantage of inherent regularity and structure of the problem. Superresolution algorithms must be robust with respect to various sources of image degradations. These include unknown sensor noise, unknown or varying camera characteristics from frame to frame, unknown modelling error, etc. Furthermore, the algorithms must be driven only by the sensor data. No detailed information about noise or camera characteristics is given.

The goal of this thesis is a complete superresolution algorithm which can be applied in realistic applications. The thesis examines superresolution under two different frameworks: an iterative approach and an interpolation-restoration approach. Fast and robust techniques are presented for various components of superresolution. We develop a projection-based framework for frame-to-frame motion estimation and image registration aspects of superresolution. For the iterative approach, the high resolution image estimate is the solution to a regularized least squares system. We propose new preconditioners to accelerate convergence for the conjugate gradient method applied to the regularized least squares problem. We also consider the issue of regularizing discrete ill-posed underdetermined problems and derive new formulas for two regularization parameter estimation techniques. Very often, the blurring operators are unknown or reliable estimates are unavailable. We develop new techniques to identify unknown blur from multiple low resolution frames. We also present a novel wavelet interpolation-restoration approach to superresolution. Numerical experimental results demonstrate the effectiveness of the proposed methods.

Acknowledgements

A few lines of acknowledgement do not fully express my gratitude and appreciation for those guided and supported me through these last five and a half years. I have been fortunate to be surrounded by teachers, family and friends with whom I can share all my joys and frustrations.

This thesis would not have been possible without the guidance of my advisors, Professors Gene Golub and Peyman Milanfar. They are my role models, inside and outside of academics. I am also in debt to Professor Michael Saunders, whose careful reading of this thesis made tremendous improvement to its content. I would also like to thank Professors Carlo Tomasi and Robert Gray for their helpful suggestions and comments.

The gang at SCCM has made it a joy to come to the office every day. I learned much from our technical discussions and value their friendships. In alphabetical order, they are Melissa, Erik, Doug, Paul H., Paul L., Ian, Joel, Matthias, GaoFeng, Ron. My appreciation also goes to Arden King and Evelyn Boughton for all their help.

To my wonderful girlfriend, Hà. You have been such a loving, caring, and supportive partner in good and bad times. I know that there is a great future waiting for us.

Ba, má, and Bill, this is for all of us.

To my parents, for all your sacrifices

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Image superresolution refers to image processing algorithms that produce high quality, high-resolution (HR) images from a set of low quality, low-resolution (LR) images. Naturally, there is always a demand for better quality images. The level of image detail is crucial for the performance of several computer vision algorithms. Target recognition, detection and identification systems are some of the military applications that require the highest quality achievable. License plate readers, surveillance monitors, and medical imaging applications are examples of civilian applications with the same requirement. In many visual applications, both civilian and military, the imaging sensors have poor resolution outputs. When resolution can not be improved by replacing sensors, either because of cost or hardware physical limits, we can resort to superresolution algorithms. Even when superior equipment is available, superresolution algorithms are an inexpensive alternative.

Multiframe superresolution, at its core, is a process by which one gains spatial resolution in return for temporal bandwidth. Temporal bandwidth refers to availability of multiple frames of the same scene. Lukosz [55, 56] was first to realize this possibility. However, superresolution can not perform miracles. Extracting subpixel information from a single frame or a sequence of identical frames is possible only if we have additional information about the scene. In this thesis, we assume no prior knowledge about the given image sequence. Hence, in order to obtain superresolution, there must be nonredundant information among the frames. We must be able to translate data temporal bandwidth into subpixel image content. Each LR frame provides a different "look" at the same scene. Theoretically, nonredundant information about the scene can be obtained by providing different lighting conditions or with different sensors. This is the multichannel data fusion superresolution problem. With just one imaging device and under the same lighting conditions, we require that there be some relative motions from frame to frame. Frame-to-frame motion can be a combination of camera platform motion relative to the scene, moving objects in the scene, and camera jitters. For example, in satellite imaging, images of the ground below are captured as the camera orbits the earth, whereas in surveillance and monitoring applications, the camera is placed on a fixed platform, and observed objects move within the scene. Motion and nonredundant information are what make superresolution possible. Armed with this information, we are able to extract subpixel content at a higher resolution than in each individual frame.

Superresolution is a computationally intensive process. Some video applications may require superresolution to be done on-the-fly; data must be processed as they are received. Imagine a superresolution integrated circuit (IC) inside the camera processing sensor data and displaying the refined results. Even for offline applications, speed is an essential requirement of a practical superresolution algorithm. The algorithm must manipulate hundreds of thousands pixel data efficiently. To that end, it must take advantage of inherent regularity and structure of the problem.

But speed should not be gained at the cost of robustness. Superresolution algorithms must be robust with respect to various sources of frame degradations. These include *unknown* sensor noise, *unknown* or *varying* camera characteristics from frame-to-frame, *unknown* modeling error, etc. Furthermore, the algorithms must be driven only by the sensor data as no information about noise or camera characteristics may be given.

The goal of this thesis is a complete superresolution algorithm which can be applied in realistic applications. We examine superresolution under two different frameworks. We propose fast and robust techniques for various stages of superresolution. Timings and quality of results will demonstrate the effectiveness of our methods.

1.1 Problem Description

In this section, we describe a straightforward and efficient model for superresolution which will serve as the foundation for the development of algorithms for the rest of the thesis. Although superresolution and multichannel data fusion can both be described with one model, for simplicity and ease of presentation, our superresolution representation will consist of LR data from only one imaging source. We assume consistent lighting conditions, negligible optical distortions, that objects observed are acquired under orthographic projections, and that individual scene motions can

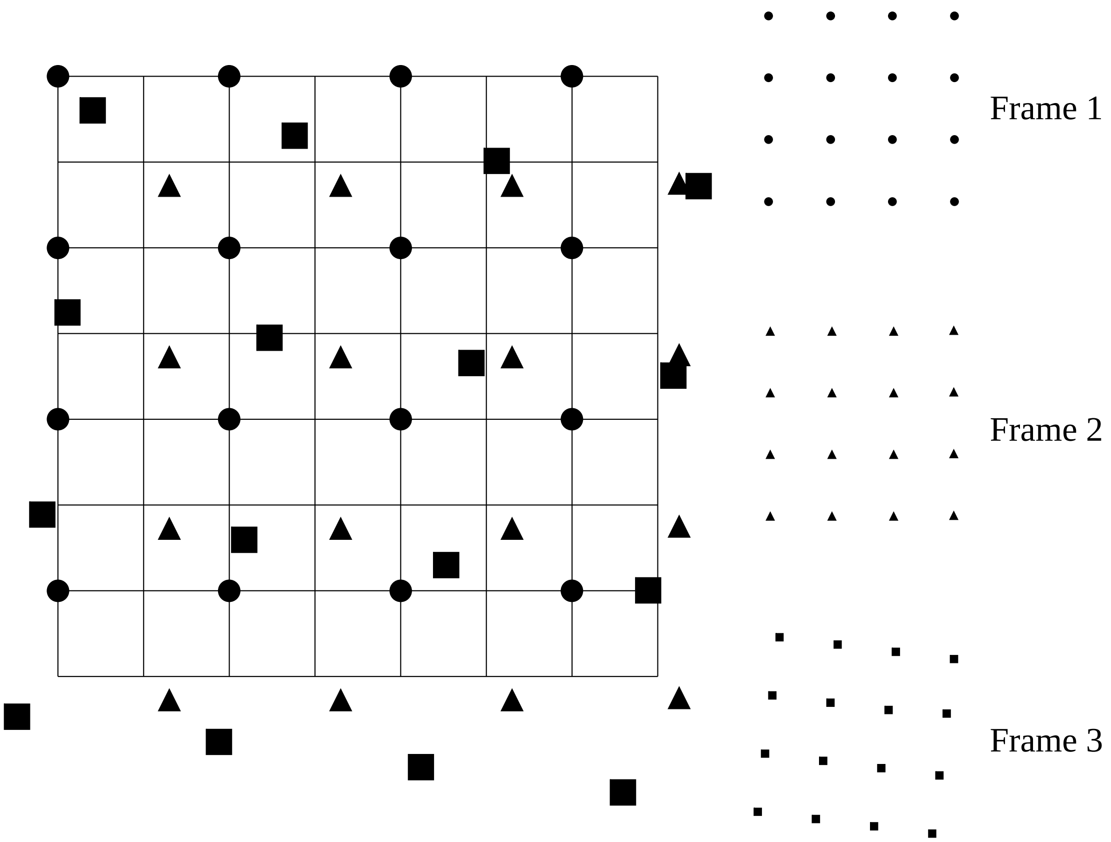


Figure 1.1: Low-resolution data on a high-resolution grid.

be modeled as affine transformations. While these simplifications result in some loss of generality, the model will be adequate for the majority of superresolution imaging applications.

The problem can be stated as follows:

> *Given a set of degraded LR frames* $\{\mathbf{f}_k\}_{k=1,\ldots,p}$ *each* $M \times N$ *pixels in dimension under the conditions above, and a desired enhancement factor* r*, reconstruct an enhanced/restored HR image with dimensions* $rM \times rN$*.*

Figure 1.1 illustrates the problem setup. The figure shows three 4×4 pixels LR frames on an 8×8 HR grid. Each symbol (square, circle, triangle) indicates

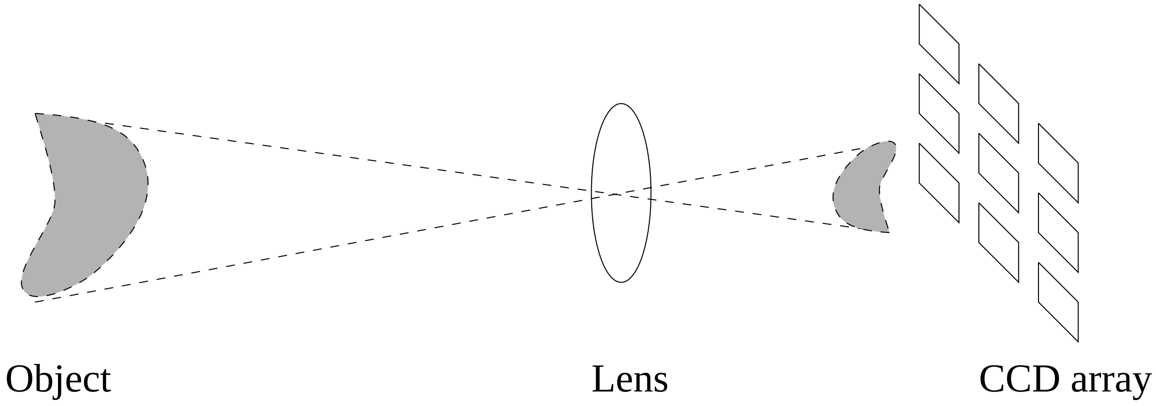


Figure 1.2: CCD camera model.

the sampling points of a frame with respect to the HR grid. We pick an arbitrary frame as a reference frame; in this case, the frame marked by the circular symbols. The sampling grid for the triangular frame is just a simple translation of the reference frame grid. The motion between the sampling grid for the square frame and the reference frame grid include translational, rotational, and magnification (zoom) components. The goal of superresolution is to interpolate and restore values at the HR grid points.

In order to solve for the unknown HR values, we first model the forward process that takes the ideal HR image to a degraded LR frame. Many imaging devices today, such as infrared and charged-coupled device (CCD) cameras, consist of arrays of light detectors. A detector determines pixel intensity values depending upon the amount of light detected from its assigned area in the scene [42]. Resolution of images produced by the camera is proportional to the density of detector array. Figure 1.2 shows a simplistic model of a CCD camera.

The camera lens produces a blurred version of the object. The CCD array turns this degraded analog signal into a discrete 2-D image with quantized intensity values. In addition, the images are contaminated by additive noise from various sources: quantization errors, sensor measurement errors, model errors, etc. Ideally, we would like to have a high density CCD array placed *in front of* the camera lens to capture an undegraded RH image, and this HR image is what we seek to reconstruct.

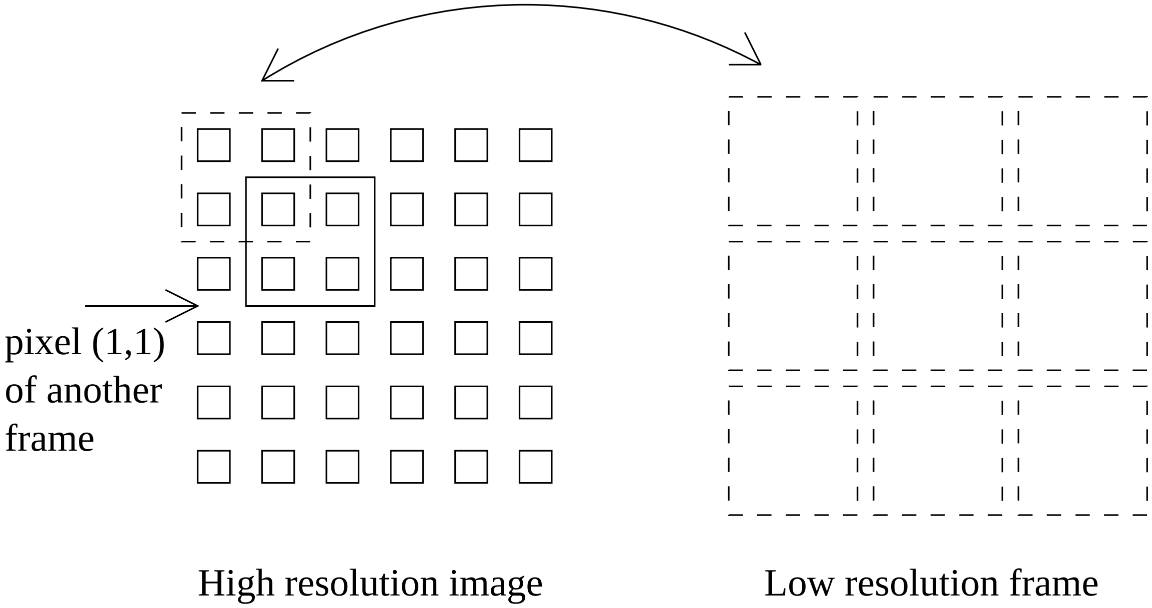


Figure 1.3: Relationship between LR and HR pixels.

Figure 1.3 illustrates the relationship between LR and HR pixels. We model each LR frame as a noisy, uniformly down-sampled version of the RH image which has been shifted and blurred. Following [24], the forward relationship between a degraded, LR frame and the ideal RH image can be described as:

$$\mathbf{f}_k = DC_kE_k\mathbf{x} + \mathbf{n}_k, \quad 1 \leq k \leq p, \tag{1.1}$$

where D is the down-sampling operator, C_k's are the blurring/averaging operators, E_k's are the affine transforms which map the HR grid coordinate system to the LR grid systems, $\mathbf{x}$ is the unknown ideal HR image, and $\mathbf{n}_k$'s are the additive noise vectors. The LR frames $\mathbf{f}_k$ are given, and the decimation operator D is known. The blurring operator C_k and camera lens characteristics are in general unknown. However, the blurring process can be well approximated to be linear spatially invariant (LSI). The scene motions for each frame relative to the reference frame are also generally not known. Finally, with multiple independent sources of error, the central limit theorem allows us to assume Gaussian normal distribution for the additive noise vectors $\mathbf{n}_k$ with possibly unknown variance. Each frame $\mathbf{f}_k$, $M \times N$ pixels in dimension, becomes a column $MN \times 1$ vector by columnwise reordering. Pixel $(1, 1)$ of the 2-D frame is ordered first, pixel $(1, 2)$ is second, and so forth. The unknown ideal image $\mathbf{x}$ is reshaped into a $r^2MN \times 1$ column by the same columnwise ordering. The matrices E_k's are square $r^2MN \times r^2MN$ matrices representing the affine transforms applied to the ideal image. The matrices C_k's are also $r^2MN \times r^2MN$ square matrices represent the blurring operators, and D is an $MN \times r^2MN$ decimation matrix. By stacking the frame equations (1.1) we get

$$\mathbf{f} = H\mathbf{x} + \mathbf{n}, \tag{1.2}$$

where $\mathbf{f}, \mathbf{n}$ are now $pMN \times 1$ vectors and H is the complete system matrix with dimensions $pMN \times r^2MN$. The shape of the system depends on the number of available nonredundant LR frames. If $p < r^2$, we have an underdetermined system. If $p = r^2$, the system will be square. And if $p > r^2$, we have an overdetermined system. All three cases are valid in practice. The matrix H is typically ill-conditioned, very large, and sparse. The dimensions of H are directly related to the number of data samples and unknowns, which are usually in the tens of thousands. Solving (1.2) is a formidable computing challenge. Yet, as we will describe, there are inherent structures and regularity which we can exploit. We have already mentioned above the spatial invariant property of the blur. Another useful property is the fact that for each frame, data points are sampled in a rectangular grid. However, if pixel values from all frames are considered together as in Figure 1.1, the data are irregularly sampled. This special case of irregular sampling is called interlaced sampling [77]. These two properties will be the basis of many techniques proposed in this thesis.

In the forward process (1.1), scene motions are applied first to the ideal image. Almost all existing superresolution algorithms estimate scene motions directly from the LR data. Unless individual scene motions are spatially invariant, e.g. purely translational, the motion estimates are corrupted by blur degradations. Only Shah

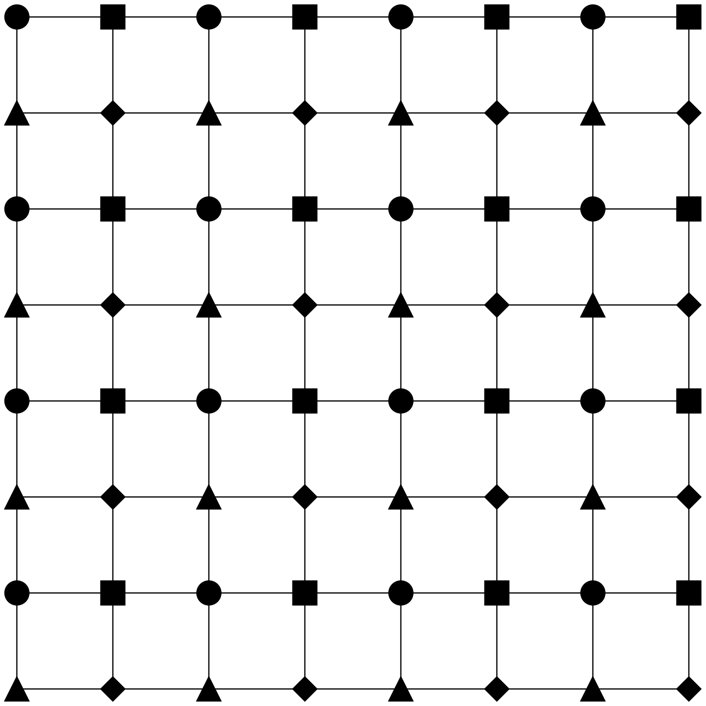


Figure 1.4: Image restoration as a special case of image superresolution.

and Zakhor [76] considered resolution enhancement with a motion registration component that accounts for the blurring and aliasing effects in the imaging process. In this thesis, we will use the customary approach of estimating motion directly from the LR frames since numerical experiments have shown that there is no significant loss of image reconstruction quality.

1.2 Related Problems

Image restoration is a special case of image superresolution. Linear restoration models have the form:

$$\mathbf{f} = C\mathbf{x} + \mathbf{n}, \tag{1.3}$$

where $\mathbf{f}$ is the noisy and blurred data image, C is the linear blurring operator, $\mathbf{x}$ is the ideal image we wish to restore, and $\mathbf{n}$ is the additive noise vector. Equation (1.3) can be rewritten as

$$\mathbf{f}_k = DCE_k\mathbf{x} + \mathbf{n}_k, \quad 1 \leq k \leq r^2, \tag{1.4}$$

where r is an arbitrary decimation factor, equivalent to the resolution enhancement factor in (1.1), the LR "frames" $\mathbf{f}_k$'s are shifted, blurred versions of the original HR image down-sampled by a factor of r, and E_k's represent the relative motion shifts covering all possible vertical and horizontal motions. Thus, image restoration can be restated as an image superresolution problem with a desired enhancement factor of r and a set of r^2 LR frames with all possible horizontal and vertical HR pixel shifts to cover the entire HR grid. Figure 1.4 illustrates this connection, where in this example, $r = 2$. The noisy, blurred image is partitioned into four LR "frames" each marked by a distinct symbol. Using the frame marked by the circle symbol as reference, the frame marked by the square symbol contains sampling points at one HR pixel shift in the horizontal direction. Similarly, the triangles mark sampling points at one HR pixel shift in the vertical direction, and the diamonds sampling points at one HR pixel shift diagonally. Image restoration is simply image superresolution with regularly sampled LR data completely covering the HR grid. The importance of this connection will be further examined a little later.

1.3 Outline of Thesis

The main goal of this thesis is the development of a complete superresolution algorithm that can be used in realistic applications with minimal operator/human supervision. The input into the algorithm is a set of LR, degraded frames, and the output is a restored HR image. The algorithm can not assume any detailed knowledge about camera characteristics such as point spread function (PSF) or noise level estimates. The algorithm must be computationally efficient to be practical for data intensive applications such as video sequence enhancement.

We examine superresolution under two frameworks. The first approach solves the superresolution equation (1.1) by a total iterative process. We consider the matrices D, C_k's, and E_k's together as a single linear operator and solve for a regularized least squares estimate by an iterative method. The connection between superresolution and restoration suggests the second approach, which is a combination of interpolation and restoration. We first interpolate the LR values at HR grid points with nonuniformly sampled data. This interpolation step reduces the superresolution problem to a deblurring/restoration problem which has been extensively studied.

With this understanding, we can decompose superresolution into various image processing components. Figure 1.5 displays the overview of the components of superresolution. The chapters in this thesis roughly correspond to the flow chart items. Chapter 2 deals with the frame-to-frame motion estimation and image registration aspects of superresolution. Both approaches to superresolution require registering the relative motions between the LR frames as a preprocessing step. For the iterative approach, the HR image estimate is the solution to a regularized least squares system. Chapter 3 describes preconditioners to accelerate convergence for the conjugate gradient (CG) method applied to the regularized least squares problem. Chapter 4 handles the issue of regularizing discrete ill-posed underdetermined problems. Very often, the blurring operators C_k's are unknown or reliable estimates are unavailable. Chapter 5 develops techniques to identify unknown blur from multiple LR frames. Our wavelet interpolation-restoration approach to superresolution is presented in Chapter 6. Finally, we have some conclusions and remarks on future directions of research in Chapter 7.

1.4 Previous Work

Superresolution reconstruction from multiple frames is a relatively new class of restoration problems. Earlier approaches to superresolution were based on extrapolating image spectrum from a single frame [47]. Various techniques were proposed,

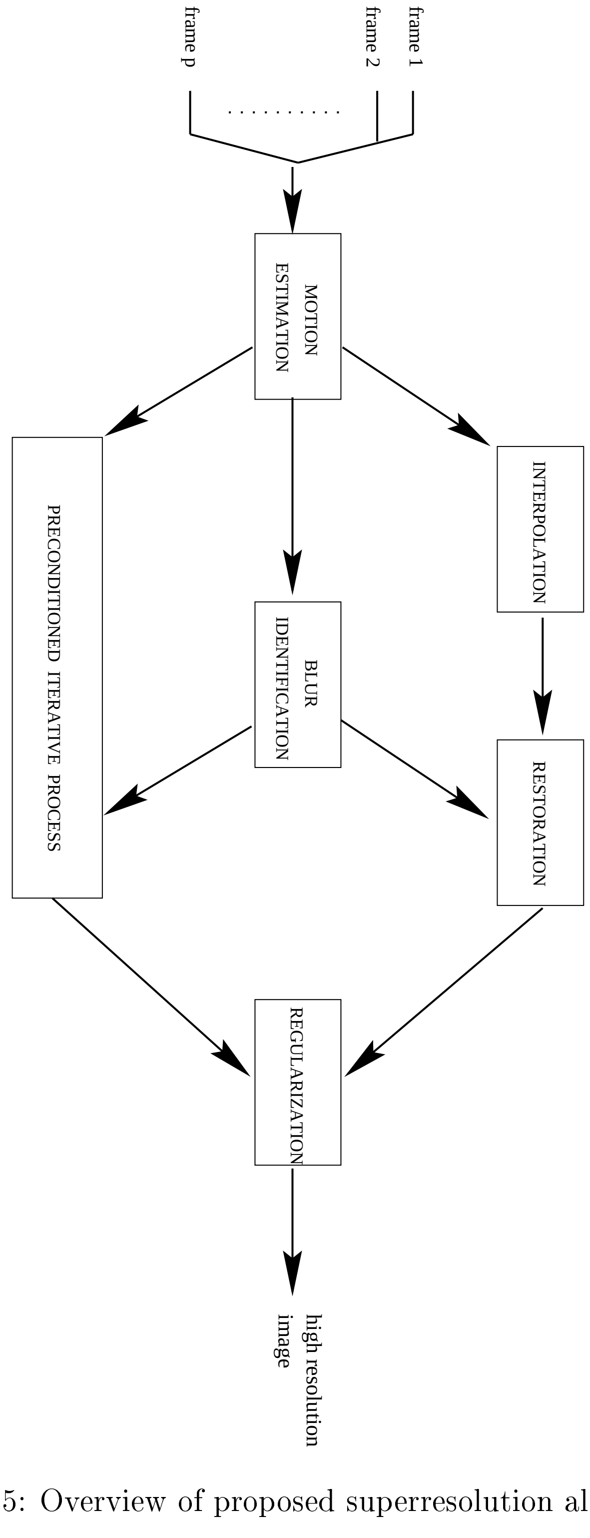


Figure 1.5: Overview of proposed superresolution algorithms.

such as analytic continuation, extrapolation via prolate spheroidal wave functions by Slepian and Pollak [80], and extrapolation by error energy reduction by Gerchberg [29] and Papoulis [68].

Tsai and Huang [87] were first to superresolve a single HR image from several down-sampled LR frames (without blur). They considered interpolation from p LR frames $\{\mathbf{f}_k\}_{k=1,\dots,p}$, each shifted from a reference frame by some shift δ_k. Frame $\mathbf{f}_k$ can be considered as samples from a continuous signal $f(x+\delta_k)$, where $f(x)$ is the ideal continuous image. In the frequency domain,

$$F_k(\omega) = e^{i\delta_k \omega} F(\omega), \tag{1.5}$$

where $F_k(\omega), F(\omega)$ are the continuous Fourier transforms of $f(x+\delta_k)$ and $f(x)$, respectively. The discrete Fourier transforms of the LR frames can be expressed as

$$\mathbf{F}_k = \Phi \mathbf{f}_k, \tag{1.6}$$

where $\Phi = \left(exp\left(-i2\pi\frac{jn}{N}\right)\right)_{j,n}$ is the discrete Fourier transform matrix. The relationship between the continuous Fourier transform and the discrete Fourier transform gives us

$$F_{kn} = \frac{1}{T}\sum_{m=-\infty}^{\infty} F_k\left(\frac{n}{NT} + m\omega_s\right), \tag{1.7}$$

where $\mathbf{F}_k = (F_{kn})_n$, T is the temporal spacing from one frame sample to the next, and ω_s is the sampling frequency of each frame. Combining (1.5), (1.6), and (1.7), they were able to calculate values of the frequency spectrum $F(\omega)$ at $-L\omega_s, -L\omega_s + \frac{1}{NT}, \dots, L\omega_s - \frac{1}{NT}$, with L being the integer such that $F(\omega) = 0$ for $|\omega| \geq L\omega_s$, resulting in an improvement of $2L$ factor in resolution.

Sauer and Allebach [73] modelled superresolution as an interpolation problem with nonuniformly sampled data. They used a projection onto convex sets (POCS) algorithm to reconstruct the unknown values:

$$\mathcal{F}^{(l+1)} = P_n \cdots P_2 P_1 \mathcal{F}^{(l)}, \tag{1.8}$$

where $\mathcal{F}^{(l)}$ is the lth approximate of the ideal HR image $\mathcal{F}$, and P_i are projection operators which correct for errors between $\mathcal{F}^{(l)}$ and $\mathcal{F}$ and impose band-limitedness constraints. The solution to the fixed point iteration (1.8) is their estimate to $\mathcal{F}$.

Aizawa *et al.* [3] also modeled superresolution as an interpolation problem with nonuniform sampling and used a formula related to Shannon's sampling theorem to estimate values on a HR grid. All [87, 3, 73] ignored the effect of sensor blurring. Tekalp *et al.* [83] later generalized Tsai and Huang's algorithm to include blurring and sensor noise and proposed the additional restoration step for the interpolation algorithms. Frieden and Aumann [28], Stark and Oskoui [81] and Irani and Peleg [46] incorporated sensor blur into a set of linear equations, which is solved by stationary iterative methods. Irani and Peleg proposed a back-projection method

$$\mathcal{F}^{(l+1)} = \mathcal{F}^{(l)} + \sum_k H_k \left(\mathbf{f}_k^{(l)} - \mathbf{f}_k \right), \tag{1.9}$$

where H_k is a normalized back-projection operator containing information about the blurring and sampling process, $\mathbf{f}_k$ is the original kth data frame, and $\mathbf{f}_k^{(l)}$ is the simulated kth frame using $\mathcal{F}^{(l)}$ as the HR image.

Subsequent work on superresolution falls into one of the three approaches described above: frequency domain, interpolation-restoration, and iterative spatial domain methods.

Under the frequency domain category, Kim *et al.* [51] used the aliasing relationship between the undersampled LR frames and a reference frame to solve the problem in the wavenumber domain using a weighted recursive least squares method. In [52], Kim and Su advanced this technique to include blurring degradations.

Ur and Gross [88] considered Papoulis' generalized multichannel sampling theorem for interpolating values on a higher resolution grid. Because the light detectors are not ideal lowpass filters, some high frequency information about the scene is represented in the image in aliased form. Papoulis' theorem reconstructs this aliased high frequency content by taking properly weighted sums of the spectral information from the LR frames. Shekarforoush and Chellappa [77] extended Papoulis' theorem for merging of nonuniform samples of multiple channels into RH data.

In a similar vein to Irani and Peleg's iterative back-projection technique, Abdou [1] employed Kaczmarz's method to iteratively refine the HR estimate. This row-projection method allows the user to process the data frames in batches. As more frames become available, the HR estimate can be updated with the new information.

Our model for the degradation process from ideal HR image to measured LR frames follows that of Elad and Feuer [25]. They form the following system of equations

$$\mathbf{f}_k \;=\; D_k C_k E_k \mathcal{F} + \mathbf{n}_k, \quad 1 \le k \le p, \tag{1.10}$$

where D_k represents the decimation operator, C_k represents the time and space varying blurring operator, E_k represents the geometric warping operator, and $\mathbf{n}_k$ is an additive noise vector. They proposed a hybrid approach that combines a maximum a posteriori (MAP) constraint with additional non-quadratic constraints

$$\begin{aligned} \hat{\mathcal{F}} &= \operatorname{argmin}_{\mathcal{F}} \left\{ \sum_k \|\mathbf{f}_k - D_k C_k E_k \mathcal{F}\|_{W_k}^2 + \beta \|S\mathcal{F}\|_V^2 \right\}, \\ &\quad \text{s.t.} \left\{ \mathcal{F} \in \mathcal{I}_j, \quad 1 \le j \le M \right\}, \end{aligned} \tag{1.11}$$

where W_k is a confidence weighting matrix related to the autocorrelation of the noise vector, S is the regularization operator, V is a weighting matrix, and β is some scaling constant. The convex sets $\mathcal{I}_j$ represent the non-quadratic constraints imposed on the HR estimate. An adaptive least squares method is used to solve for the minimizer of (1.11).

More sophisticated models of superresolution have appeared in the engineering literature recently. Hardie *et al.* [39] proposed a joint MAP registration and restoration algorithm using a Gibbs image prior. Schultz and Stevenson [75] used a Huber-Markov random field model with Gibbs prior to better represent image discontinuities, such as transitions across sharp edges. Patti *et al.* [69] extended the superresolution model to include nonzero aperture time (motion blur) effect.

However, previous works have not addressed the computational challenges of superresolution adequately. Regularization and other robustness issues are dealt with in an ad hoc manner. These challenges are the motivations for this thesis.

1.5 Contributions of Thesis

The main contribution of this thesis is a complete, fast, and robust image superresolution algorithm. We have constructed a modular framework for superresolution (see Figure 1.5). For each stage of the algorithm, we develop techniques with potential for real time practical applications.

- In Chapter 2, we present the projection-based approach to motion estimation. By using properties of the Radon transform, this approach reduces the 2-D motion estimation problem to two 1-D problems, lowering the computational burden significantly. New motion estimation methods can be derived from existing 2-D motion estimation methods. We show bounds on loss of accuracy under the projection framework. Motion estimation timing and accuracy results for several simulated and real image sequences are presented. We also examine the effect of registration accuracy on overall superresolution reconstruction quality. Superresolution reconstruction results under various levels of misregistration are presented.

- In Chapter 3, we describe two new preconditioners for accelerating CG convergence for superresolution. These preconditioners are based on the degradation model for each LR frame and thus, are applicable for all cases of data availability: underdetermined, square, and overdetermined. They are inexpensive to construct and use. We show bounds on the number of preconditioned iterations needed for exact convergence. We also show timing and iteration count results to demonstrate the effectiveness of our preconditioners.

- Chapter 4 deals with regularizing discrete ill-posed problems. Although there is extensive literature on regularizing *overdetermined* and *square* systems, very little has been written on *underdetermined* systems. In this chapter, we examine regularization and parameter selection for underdetermined systems. We derive new formulas for two regularization parameter criteria. We employ and compare these criteria for computing regularization parameters for superresolution.

- In Chapter 5, we consider the blur identification and the related blind superresolution problems. While the blur identification problem has been well studied, fast and robust techniques are lacking. We propose new quadrature-based methods for the blur identification problem using generalized cross-validation and demonstrate their effectiveness with blind superresolution and blind restoration results.

- In Chapter 6, we describe a fast wavelet interpolation method for interlaced sampling. This method is well-suited as an interpolation technique for superresolution data, which come in the form of interlaced sampling. We present wavelet superresolution results from applying a combination of this interpolation method with a restoration step.

Although algorithms in this thesis are developed with superresolution in mind, they may be useful in other settings. We will point out other potential applications along the way.

Chapter 2

Motion Estimation

2.1 Introduction

Superresolution from multiple frames is possible only in the presence of motion. In some superresolution applications, such as satellite remote sensing, relative motion from frame to frame, based on the satellite orbiting velocity and path, is known beforehand at least approximately. For others, motion must be estimated from raw data as a preprocessing step to superresolution. In these cases, motion estimation is an essential component of superresolution. It is also an important tool in other imaging applications including image stabilization, video compression using motion compensation, and motion detection.

Frame-to-frame motions provide different views of the scenes or objects of interest. Frame motions may combine a global motion component as a result of camera movement and local scene motions as observed objects move within the scene. If these frame motions are sufficiently smooth they may be approximated by an affine model locally. Although this model is not accurate for more complex movements and large image regions where there may be multiple moving objects, it serves as a good approximation for smaller regions. For recent work on the piecewise affine model for image motion, we refer the reader to Wang and Adelson [91] and Ju *et al.* [48]. In this thesis, we assume that a single set of affine parameters will suffice to describe the motion from one frame to the next. Our image registration framework has many similarities with optical flow calculation where a dense motion flow field is computed for each LR frame. For optical flow calculation, a single motion vector representing purely translational motion is computed for each region of the image. These regions can be quite small in size (down to one pixel) whereas for motion estimation in image superresolution, we are interested in resolving rigid moving objects occupying a larger portion of the image. We refer the reader to the survey on optical flow techniques by Barron *et al.* [8] for further details.

In order to extract subpixel information content from the sequence, each frame must be registered accurately with respect to some reference to within subpixel accuracy. Intuition dictates that in order to achieve an order n resolution enhancement, image registration must be accurate to within $\frac{1}{n}$ pixel. As we shall see, superresolution results from misregistered and misaligned frames are visually unsatisfactory.

In this chapter, we first describe the affine gradient-based model for motion estimation. We review two approaches for computing motion. The first technique employs standard least squares estimation in the image domain to find the parameters of motion. The second uses plane-fitting algorithms in the frequency domain. In the second part of the chapter, exploiting properties of the Radon transform, we propose an efficient gradient-based projection framework. This framework generates a new class of motion estimation techniques for the 2-D motion estimation problem by reducing it to two independent 1-D problems. We give an upper bound for potential performance loss as a result of this projection step. We present numerical experimental results verifying our claims that our projected motion estimation approach is faster, without significant loss of accuracy. We also examine the effect of registration errors on superresolution reconstruction results.

2.2 Preliminaries

Motion estimation is a common area of interest for many different computer vision applications. The result is a large body of research with numerous proposed techniques. Almost all of them can be categorized under one of four groups: correlation or block-matching, feature-based, gradient-based, and spatiotemporal filtering [41]. We concentrate our attention on gradient-based techniques in this chapter. For a comprehensive review on the topic, the reader is referred to the survey by Brown [14].

2.2.1 Intensity Conservation Assumption

A fundamental assumption underlying gradient-based methods is that the total intensity is conserved from frame to frame. This assumption holds under fairly ideal conditions, e.g. no large motions, consistent lighting conditions, and no occlusions. Let $f(x, y, t)$ be the image intensity function in space and time. Then we can express intensity conservation as follows:

$$f(x, y, t) \quad = \quad f(x + \delta_1(x, y)\Delta t, y + \delta_2(x, y)\Delta t, t + \Delta t) \tag{2.1}$$

where

$$\begin{aligned} \delta_1(x, y) &= p_1 x + p_2 y + p_3 \\ \delta_2(x, y) &= p_4 x + p_5 y + p_6 \end{aligned} \tag{2.2}$$

represent the affine motion flow field about the origin $(0, 0)$. By Taylor's expansion, the first order approximation of the right hand side of (2.1) is

$$f(x + \delta_1(x, y)\Delta t, y + \delta_2(x, y)\Delta t, t + \Delta t) \approx f(x, y, t) + \nabla f(x, y, t)^T \delta, \tag{2.3}$$

where $\delta = \begin{pmatrix} \delta_1(x, y) \\ \delta_2(x, y) \\ 1 \end{pmatrix} \Delta t$. Substituting into (2.1), we obtain the gradient constraint equation:

$$\nabla f(x, y, t)^T \delta \quad = \quad 0. \tag{2.4}$$

The gradient constraint equation (2.4) was first proposed by Horn and Schunk [43] for computing optical flow fields and is the foundation of many optical flow and image registration algorithms.

2.2.2 The Aperture Problem

We call attention to the fact that components of motion to $\nabla f(x, y, t)$ can not be computed from (2.4). This is an example fo the aperture problem, which refers to situations in which the correct motion vectors can not be determined due to insufficient information. There might not be enough information in a region of interest (aperture) to solve for the correct motion vector. Consider Figure 2.1 for example. The diagonal arrows indicate the true motion vector which coincides with

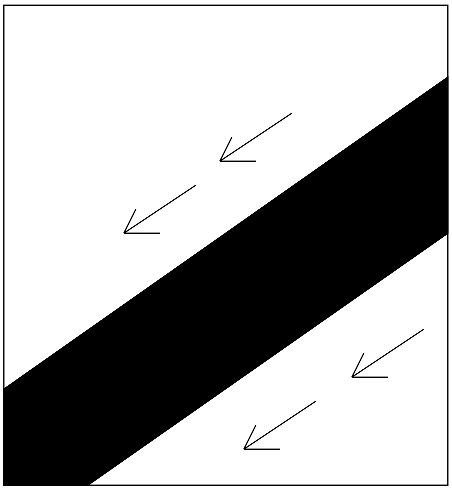


Figure 2.1: Aperture problem: insufficient information to determine true motion vector

the orientation of the filled-in region. Because of the one-dimensional structure of the scene, the magnitude of the true motion vector is ambiguous. We need additional constraints or information about the motion in order to compute the motion vector correctly.

2.3 Motion Estimation Methods

We introduce in this section two popular motion estimation methods, the least squares estimate and the frequency-domain, plane-fitting method. Most superresolution algorithms either assume that motion parameters are known or use the gradient-based least squares estimate of motion. If the affine model describes frame-to-frame motion well, gradient-based approaches are very accurate since only six parameters are needed. The second method, operating in Fourier domain, is also based on the intensity conservation assumption. The method is derived from the interpretation of the conservation assumption in the frequency domain.

2.3.1 Least Squares Estimate

Let us rewrite (2.2) in matrix form,

$$\begin{bmatrix} \delta_1(x,y) \\ \delta_2(x,y) \end{bmatrix} = A\mathbf{p}, \tag{2.5}$$

$$A = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{pmatrix}, \quad \mathbf{p} = \begin{pmatrix} p_1 & \cdots & p_6 \end{pmatrix}^T. \tag{2.6}$$

The gradient constraint equation (2.4) in matrix form becomes

$$f_s^T A\mathbf{p} + f_t = 0, \tag{2.7}$$

with $f_s = (f_x \quad f_y)^T$. Summing over all pixels in the region of interest, the regularized least squares estimate of motion is the minimizer of

$$\operatorname{argmin}_{\mathbf{p}} \sum_{x,y} \left(f_s^T A\mathbf{p} + f_t\right)^2 + \lambda\|\mathbf{p}\|_2^2, \tag{2.8}$$

for some regularizing scalar λ. The purpose of the regularization term in (2.8) is to encourage "well-behaved" solutions. With $\lambda = 0$, we recover the least squares solution. Other regularization approaches, e.g., robust estimation framework [13], have also been proposed. Differentiating with respect to $\mathbf{p}$ and setting to 0, we get

$$\sum_{x,y} \left(A^T f_s f_s^T A\mathbf{p} + A^T f_s f_t\right) + \lambda\mathbf{p} = 0. \tag{2.9}$$

Hence,

$$\mathbf{p} = -\left(\sum_{x,y} A^T f_s f_s^T A + \lambda I\right)^{-1} A^T f_s f_t. \tag{2.10}$$

Operator	Horizontal	Vertical
Roberts	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
Smoothed	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Isotropic	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

Table 2.1: Some common gradient operators.

An important issue in this approach is how to compute the numerical partial derivatives f_x, f_y, and f_t at the pixel grid points. The spatial derivatives f_x, f_y can be computed by filtering the image with a gradient operator. Table 2.1 lists a few commonly used gradient operators (cf. [47, p. 349]). To approximate f_t, Irani and collaborators [46, 50] and others [39] used the intensity difference between two consecutive frames, $f(x, y, t_0)$ and $f(x, y, t_0 + 1)$. This is equivalent to applying a simple mask $\begin{bmatrix} 1 & -1 \end{bmatrix}$ in the temporal axis. If the motion is smooth, a natural generalization of this process would be to better approximate the temporal partial derivative using more sophisticated masks.

2.3.2 Frequency-domain Plane-fitting Method

Uniform translating motion in the image domain has an interesting interpretation in the Fourier domain. If the image sequence is moving uniformly with motion vector (p_3, p_6), then in the Fourier domain, the energy of the image sequence considered as a 3-D spatiotemporal function will be concentrated on a plane through the origin whose orientation is related to the direction of motion. To see this, let $x' = x +$

$\delta_1(x,y)\Delta t,\;\; y' = y + \delta_2(x,y)\Delta t,\;\; t' = t + \Delta t$. Equivalently,

$$\begin{bmatrix} x' \\ y' \\ t' \end{bmatrix} = \begin{bmatrix} 1+p_1\Delta t & p_2\Delta t & 0 \\ p_4\Delta t & 1+p_5\Delta t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ t \end{bmatrix} + \begin{bmatrix} p_3\Delta t \\ p_6\Delta t \\ \Delta t \end{bmatrix}, \qquad (2.11)$$

or in vector form,

$$\mathbf{s}' = M\mathbf{s} + \mathbf{d}. \qquad (2.12)$$

Let $F(\omega_x, \omega_y, \omega_t)$ denote the Fourier transform of $f(x,y,t)$. Apply the Fourier transform to both sides of (2.1) to get

$$F(\omega_x, \omega_y, \omega_t) = \int\int\int_{\mathcal{D}} f(x', y', t') e^{-i2\pi(\omega_x x + \omega_y y + \omega_t t)} dx dy dt. \qquad (2.13)$$

Rewriting in vector notation

$$F(\mathbf{w}) \;=\; \int_{\mathcal{D}} f(\mathbf{s}') e^{-i2\pi \mathbf{w}^T \mathbf{s}} d\mathbf{s}. \qquad (2.14)$$

Note that $dx'dy'dt' = |\det(M)|\, dxdydt$. The term $\det(M)$, determinant of M, represents the magnification factor of the transformation. Using (2.12) we have

$$\begin{aligned} F(\mathbf{w}) &= \frac{1}{|\det(M)|} \int_{\mathcal{D}} f(\mathbf{s}') e^{-i2\pi \mathbf{w}^T M^{-1}(\mathbf{s}' - \mathbf{d})} d\mathbf{s}' & (2.15) \\ &= \frac{e^{i2\pi \mathbf{w}^T M^{-1}\mathbf{d}}}{|\det(M)|} F(M^{-T}\mathbf{w}). & (2.16) \end{aligned}$$

When the motion is purely translational, i.e., $M = I$, the equation above becomes

$$F(\mathbf{w}) \;=\; e^{i2\pi(p_3\omega_x \Delta t + p_6 \omega_y \Delta t + \omega_t \Delta t)} F(\mathbf{w}), \qquad (2.17)$$

which is equivalent to

$$p_3\omega_x + p_6\omega_y + \omega_t = 0. \qquad (2.18)$$

Equation (2.18) represents a plane through the origin in the frequency domain with orientation directly related to the motion vector. The motion parameters can be calculated by finding the plane with highest energy in the frequency domain.

2.4 Motion Estimation through Radon Transform

We can apply a projection-based framework to the estimation processes above [57]. Define the Radon transform of the image $f(x,y)$ in the spatial scanning direction θ as follows:

$$\begin{aligned} g(z,\theta) &= \mathcal{R}_\theta f(x,y) \\ &= \int\int_{\mathcal{D}} f(x,y)\delta(z - x\cos(\theta) - y\sin(\theta))dxdy. \end{aligned} \tag{2.19}$$

Recall the notation from Subsection 2.2.1:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1+p_1 & p_2 \\ p_4 & 1+p_5 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p_3 \\ p_6 \end{bmatrix}, \tag{2.20}$$

or in compact form,

$$\mathbf{s}' = K\mathbf{s} + \mathbf{d}. \tag{2.21}$$

Let $\mathbf{l}_\theta = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}$ and consider the Radon transform of the warped image $f(\mathbf{s}')$

$$\mathcal{R}_\theta f(\mathbf{s}') = \int_{\mathcal{D}} f(\mathbf{s}')\delta(z - \mathbf{l}_\theta^T \mathbf{s})d\mathbf{s} \tag{2.22}$$

$$= \frac{1}{|\det(K)|}\int_{\mathcal{D}} f(\mathbf{s}')\delta(z - \mathbf{l}_\theta^T K^{-1}(\mathbf{s}' - \mathbf{d}))d\mathbf{s}'. \tag{2.23}$$

Let $z' = z + \mathbf{l}_\theta^T K^{-1}\mathbf{d}$ and $\mathbf{k}_\theta = K^{-T}\mathbf{l}_\theta$. Then

$$\mathcal{R}_\theta f(\mathbf{s}') = \frac{1}{|\det(K)|}\int_{\mathcal{D}} f(\mathbf{s}')\delta(z' - \mathbf{k}_\theta^T \mathbf{s}')d\mathbf{s}' \tag{2.24}$$

$$= \frac{1}{||\mathbf{k}_\theta||_2|\det(K)|}\int_{\mathcal{D}} f(\mathbf{s}')\delta\left(\frac{z'}{||\mathbf{k}_\theta||_2} - \left(\frac{\mathbf{k}_\theta}{||\mathbf{k}_\theta||_2}\right)^T \mathbf{s}'\right) d\mathbf{s}' \tag{2.25}$$

$$= \frac{1}{||\mathbf{k}_\theta||_2|\det(K)|} g\left(\frac{z'}{||\mathbf{k}_\theta||_2}, \theta'\right), \quad \theta' = \tan^{-1}\left(\frac{\mathbf{k}_\theta(2)}{\mathbf{k}_\theta(1)}\right). \tag{2.26}$$

For the special case of purely translational motion, i.e., $K = I$, the equation above reduces to

$$\mathcal{R}_\theta f(\mathbf{s}', t) = g(z', \theta, t), \quad z' = z + p_3\cos(\theta) + p_6\sin(\theta). \tag{2.27}$$

This implies that as the function $f(x, y, t)$ undergoes translational motion with velocity (p_3, p_6) in the image domain, its Radon transform $g(z, \theta, t)$ shifts by $p_3\cos(\theta) + p_6\sin(\theta)$ in the projection domain. Using two different angles of projections, we can solve for the unknown parameters p_3 and p_6. Thus, a 2-D translational motion estimation problem can be reduced to two independent 1-D problems. Since this transformation is based on special properties of the Radon transform and is independent of estimation techniques, we can develop new and faster motion estimation algorithms from existing algorithms. Namely, we can consider projected versions of the least squares estimate described in Section 2.3.1 and the frequency-domain, plane-fitting method outlined in Section 2.3.2.

The projected least squares method first decomposes the 2-D motion problem into the horizontal and vertical components, i.e. with $\theta = 0$, 90 degrees. We then apply the 1-D least squares approach to each component separately, assuming the intensity conservation holds in the projections, which, as Milanfar has shown in [64], is implied by assuming intensity conservation in the image. Milanfar [63] originally proposed the projected frequency domain algorithm. Similar to the projected least squares method, the algorithm decomposes the problem into its horizontal and vertical motion components. For each subproblem, the 2-D Fourier transform is applied to the projected sequence. A line fitting algorithm (e.g. Hough transform [23], SLIDE [2]) is then applied to the projected image spectrum to find the best fitting lines. The slopes of the two estimated lines then define the motion vector.

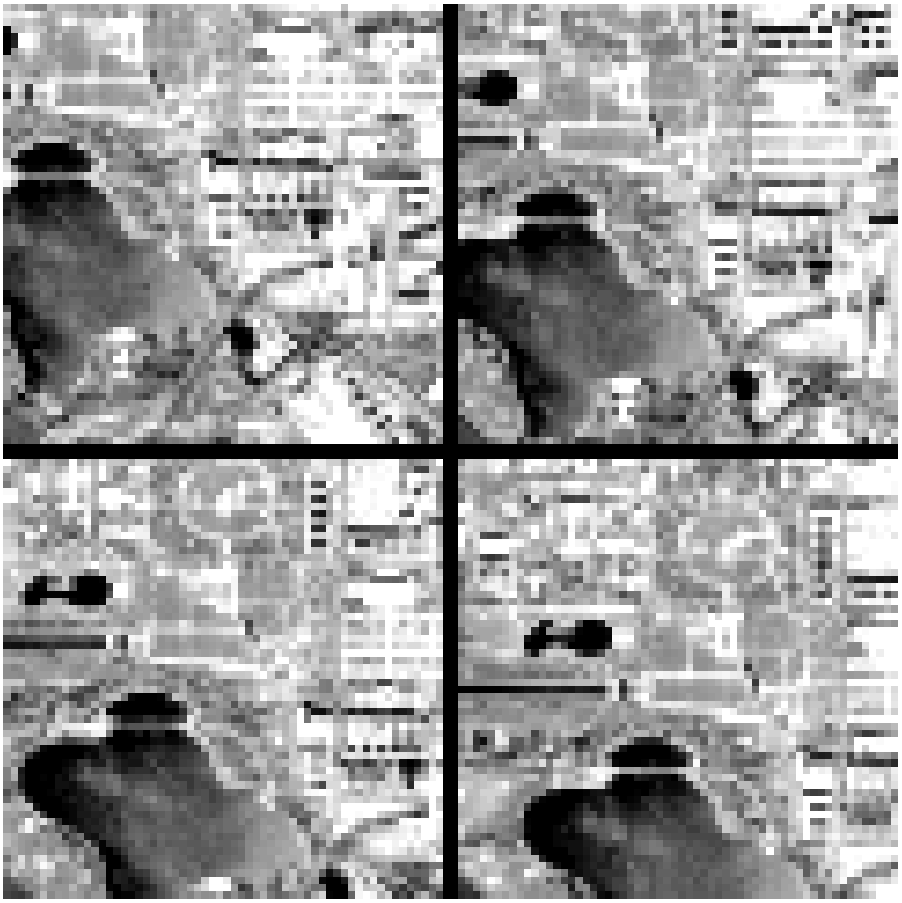


Figure 2.2: Images from Washington DC sequence, frames 1, 14, 27, 40.

2.5 Motion Estimation Experiments

We tested the accuracy and timing of the four methods described above: least squares estimate, frequency domain method, and their projected versions. We ran tests on 3 image sequences. The first sequence of 40 images is a simulated overhead view of Washington DC. Figure 2.2 shows frames 1, 14, 27, and 40 from the sequence. Figure 2.3 contains the two projected images with the directions of projection parallel to the x- and y-axis, respectively. The slopes of the line patterns are the desired velocity components. Each image is 60×60 pixels in size. The exact motion is known to be (0.5,-0.5) pixel per frame. The results presented in Table 2.2 are

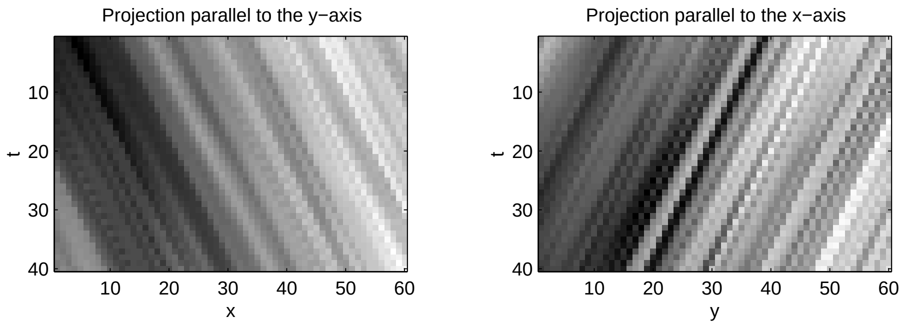


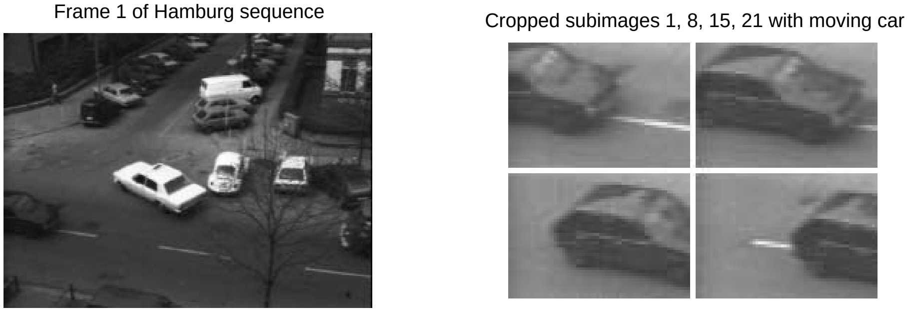
Figure 2.3: Washington DC image sequence projected onto the x- and y-axes.

compiled by averaging computed velocities for frames 21 to 30. Since exact motion is known, we show the error results in fractions of pixels. In this experiment, the LS methods outperform the FD ones. As expected, there is some loss of accuracy when the methods are applied to the projected image sequence.

The second test sequence is the well-known Hamburg sequence. This sequence contains 21 frames each of size 190 × 256. We crop subimages of size 44 × 64 containing a moving car. Ground truth velocity for this object is approximately (3,-0.5) pixels. We computed velocity vectors for frame 6 to frame 15 of the sequence. Since

Method	Mean Error (pixels)	Std Error
LS	(0.019,0.013)	(0.008,0.008)
Projected LS	(0.027,0.025)	(0.030,0.003)
FD	(0.028,0.017)	(0.017,0.000)
Projected FD	(0.048,0.041)	(0.041,0.017)

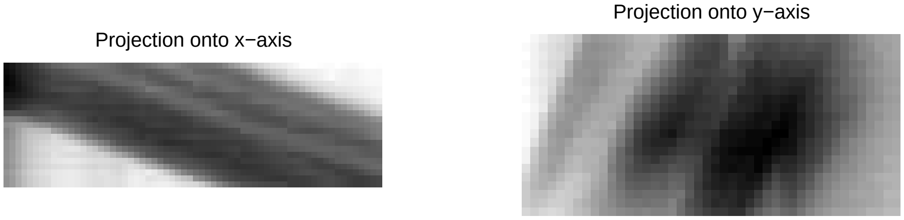
Table 2.2: Velocity errors (pixels) for Washington DC sequence for the least squares (LS) and Fourier-domain (FD) methods and their projected versions.



Frame 1 of Hamburg sequence

Cropped subimages 1, 8, 15, 21 with moving car

Figure 2.4: Hamburg sequence with cropped subimages.



Projection onto x−axis

Projection onto y−axis

Figure 2.5: Hamburg image sequence projected onto the x- and y-axes.

ground truth motion values are not known exactly, we compute the mean square error (in percent) between between consecutive frames, adjusted by the computed motion. The mean square error (in percent) is defined as

$$MSE = 100\frac{\|a\hat{h}_1 - h_2\|_F^2}{\|h_2\|_F^2}, \tag{2.28}$$

$$a = \frac{\sum_{k,l} h_2(k,l)\hat{h}_1(k,l)}{\|\hat{h}_1\|_F^2}, \tag{2.29}$$

where h_1, h_2 denotes consecutive frames, $\hat{h}_1$ the first frame warped by the computed motion vector, and $\|\cdot\|_F$ the usual Frobenius norm. Table 2.3 tabulates the estimation error for each of the methods for the Hamburg sequence. As in the first experiment, the LS methods are more accurate compared to the FD ones, and the Radon transformation only slightly degrades accuracy measure.

The third test sequence consists of 50 frames of an aerial video the Pittsburgh area. Each frame is 60×64 pixels in size. As we can see from the projection images

Frame k	LS	Projected LS	FD	Projected FD
6	0.671	0.783	1.89	3.42
7	0.587	0.742	1.90	3.38
8	0.618	0.740	6.13	2.67
9	0.755	0.767	6.03	3.51
10	0.868	0.908	4.39	2.65
11	0.662	0.659	3.62	1.98
12	0.622	0.668	2.13	2.02
13	0.680	0.804	1.70	1.50
14	0.695	0.783	2.15	1.89
15	0.549	0.633	1.67	2.78

Table 2.3: Hamburg image sequence: mean square error (in percent) between kth frame and interpolated $(k-1)$th frame using estimated motion.

Figure 2.6: Images from Pittsburgh sequence, frames 1, 17, 33, 50.

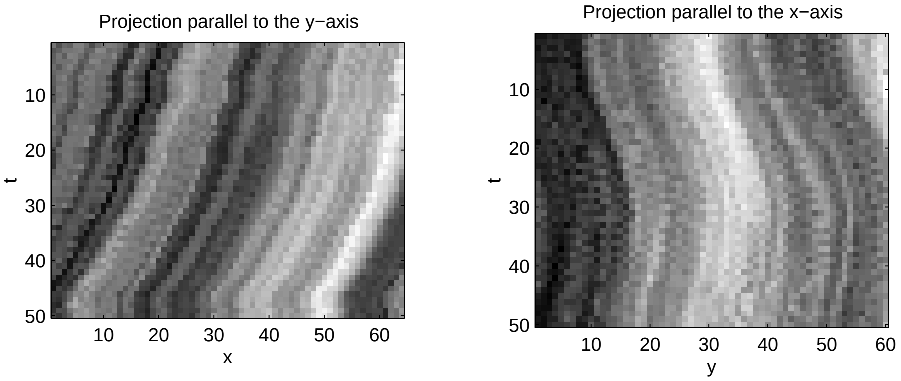


Figure 2.7: Pittsburgh image sequence projected onto the x- and y-axes.

Frame	LS	Projected LS	FD	Projected FD
21	1.32	1.35	2.20	2.06
22	4.51	4.17	4.51	4.51
23	3.63	3.73	3.96	3.96
24	4.37	4.75	4.50	4.61
25	3.92	3.98	3.99	4.11
26	1.77	1.90	2.78	2.78
27	3.68	3.49	4.56	4.56
28	2.78	2.77	3.50	3.08
29	4.45	4.50	4.44	4.57
30	4.38	4.33	4.61	4.35

Table 2.4: Pittsburgh image sequence: mean square error (in percent) between kth frame and interpolated $(k-1)$th frame using estimated motion.

(see Figure 2.7), the motion occuring in this sequence is not purely translational. However, this motion can be approximated fairly well locally with just a translational motion vector. We ran the same procedure as in the Hamburg sequence. Table 2.4 shows the computation errors for each method applied to this Pittsburgh sequence. For this test sequence, comparable results are obtained by all methods.

For all test cases, the least squares methods tend to be more accurate than frequency-domain ones. This makes sense, intuitively: Both approaches are derived from the same underlying intensity conservation assumption. However, the LS methods calculate motion parameters more directly from pixel intensity values, whereas, the FD methods only use information in the Fourier domain. Furthermore, it was shown by the experiments that projection-based methods are only slightly less accurate than their non-projected counterparts. In the next section, we discuss in more detail bounds on the loss of accuracy when using the Radon transformation.

2.6 Performance Analysis

We would like to assess the performance of the projection-based methods against the original methods under high-SNR and small-motion assumptions. We summarize here the discussion in Milanfar's paper [63], which showed a bound on the relative performance loss of projected-based approaches. We consider the case in which the frames are corrupted by Gaussian white noise:

$$u(x,y,t) = f(x,y,t) + w(x,y,t), \tag{2.30}$$

where $w(x,y,t)$ is spatiotemporally white noise with zero mean and variance σ^2. Projections of these noisy frames along the x- and y-axes are

$$p_u(x,t) = p(x,t) + w_1(x,t), \quad q_u(y,t) = q(y,t) + w_2(y,t) \tag{2.31}$$

where w_1 and w_2 Gaussian white noise with zero mean and variance $N\sigma^2$, N is the number of pixels along each side of the image. For high SNR and small motions, the approximate error covariance matrix for the velocity estimation problem is

$$Q = \sigma^2 \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}^{-1}, \tag{2.32}$$

where

$$\begin{aligned} D_{xx} &= \sum_{x,y,t} \left(\frac{\partial f}{\partial x} \right)^2, && (2.33) \\ D_{xy} &= \sum_{x,y,t} \frac{\partial f}{\partial x} \frac{\partial f}{\partial y}, && (2.34) \\ D_{yy} &= \sum_{x,y,t} \left(\frac{\partial f}{\partial y} \right)^2. && (2.35) \end{aligned}$$

For comparison with the 1-D case, we use the trace of Q,

$$C_3 = tr(Q) = \sigma^2 \left(\frac{D_{xx} + D_{yy}}{D_{xx} D_{yy} - D_{xy}^2} \right). \tag{2.36}$$

Similarly, we define the sum of variances of motion estimates from projections as a measure of 1-D performance:

$$C_2 = N\sigma^2 \left(\frac{1}{d_{xx}} + \frac{1}{d_{yy}} \right), \tag{2.37}$$

where

$$\begin{aligned} d_{xx} &= \sum_{x,t} \left(\frac{\partial p}{\partial x} \right)^2, && (2.38) \\ d_{yy} &= \sum_{y,t} \left(\frac{\partial q}{\partial y} \right)^2. && (2.39) \end{aligned}$$

The relative performance loss is bounded by the following:

$$\frac{C_3 - C_2}{C_3} \leq \frac{D_{xy}^2}{D_{xx} D_{yy}}, \tag{2.40}$$

Sequence	Covariance Bound
Washington DC	7.8352e-04
Hamburg	0.0025
Pittsburgh	2.1235e-04

Table 2.5: Covariance bound

Method	Washington DC	Hamburg	Pittsburgh
LS	4.26	12.1	5.83
Projected LS	0.932	1.22	1.53
FD	797	355	1201
Projected FD	2.44	2.35	3.76

Table 2.6: Timing results (seconds) on a Sun UltraSparc I Model 170E workstation.

where the right hand side is the correlation coefficient between the image gradients in the x and y directions.

For each test sequence, we compute the right hand side of (2.40) to determine relative performance degradation. Table 2.5 shows these performance bounds on our test sequences. These bounds are tight, agreeing with results from Tables 2.2, 2.3, and 2.4 that the projected versions are comparable to the original methods in accuracy.

Next, we demonstrate the computational advantage of the projected versions. Table 2.6 shows that the computing timings of the projected methods are significantly less than those of the original methods. The projected version of the LS methods is by far the fastest of all methods. To further illustrate our point, Tables 2.7 and 2.8 display the computational complexity analysis of each method for estimating translational motion for a sequence of p frames of $N \times N$ pixels, with

Complexity	Projected LS	LS
Projection	pN^2	0
Gradient computation	$\mathcal{O}(pN)$	$\mathcal{O}(pN^2)$
Velocity computation	$\mathcal{O}(pN)$	$\mathcal{O}(pN^2)$

Table 2.7: Computational complexity for LS methods

Complexity	Projected FD	FD
Projection	pN^2	0
FFT	$\mathcal{O}(pN\text{log}N)$	$\mathcal{O}(pN^2\text{log}N)$
Velocity computation	$\mathcal{O}(pN)$	$\mathcal{O}(pN^2)$

Table 2.8: Computational complexity for FD methods

$p < N$. The computational burden of projection work needed for projected algorithms is easily outweighed by the extra burden of computing for FFT and gradient for the non-projected versions.

2.7 Superresolution Experiments

In this section, we examine some qualitative and quantitative effects of misregistration on superresolution. We start with an image of the Stanford quad shown in Figure 2.9. We generate LR frames by applying various multiple HR pixels shift in both vertical and horizontal directions, 4×4 pixel Gaussian blur with variance 1, and down-sampling by a factor of 4 in each dimension. As a result, each LR frame has 1/16 the number of pixels as the original HR image. In the experiments below, we examine the effects of misregistration on superresolution. We simulate registration error by randomly perturbing the exact frame-to-frame motion measurements. Three cases of data availability were considered: superresolution given 10 (underdetermined), 16 (square), and 20 (overdetermined) LR data frames. Figure 2.8 shows superresolution reconstruction error as a function of registration error. The graph demonstrates a clear intuitive relationship between MSE reconstruction errors, registration errors, and amount of available data. Reconstruction quality degrades as misregistration increases. Furthermore, registration errors have more impact on reconstruction with fewer available frames.

Subjectively, reconstruction quality degrades much more rapidly than quantitative MSE measurements suggest. Figure 2.10 displays superresolution results with 10 frames and various levels of misregistration. Naturally, as the registration error increases, reconstruction quality degrades. The reconstructed image becomes

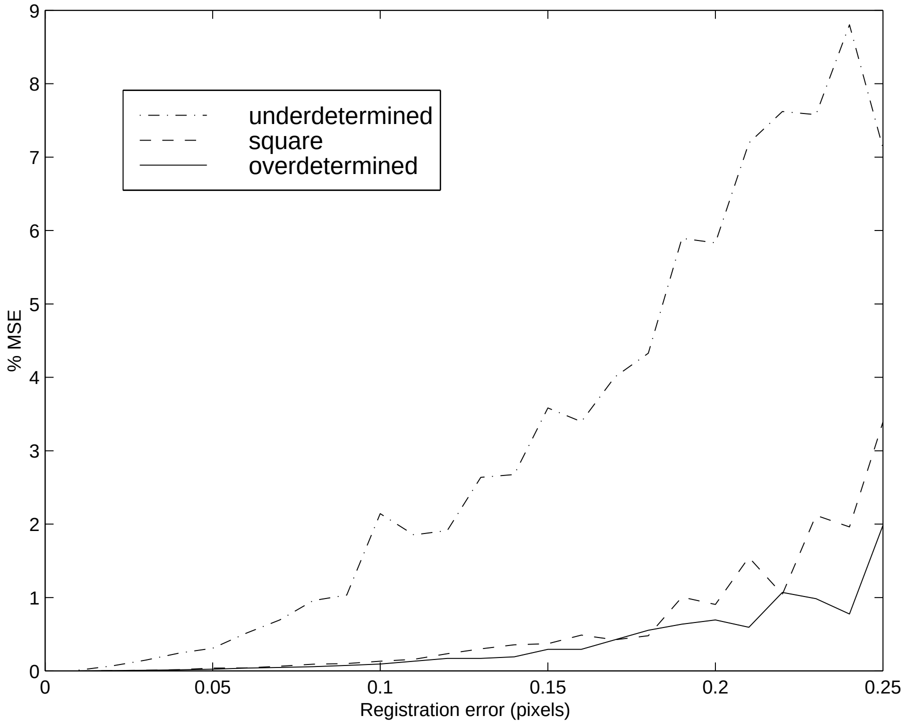


Figure 2.8: Superresolution error in the presence of misregistration.

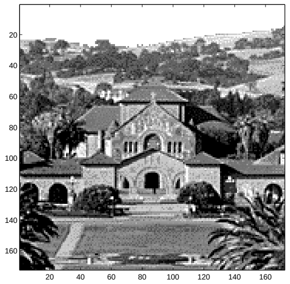
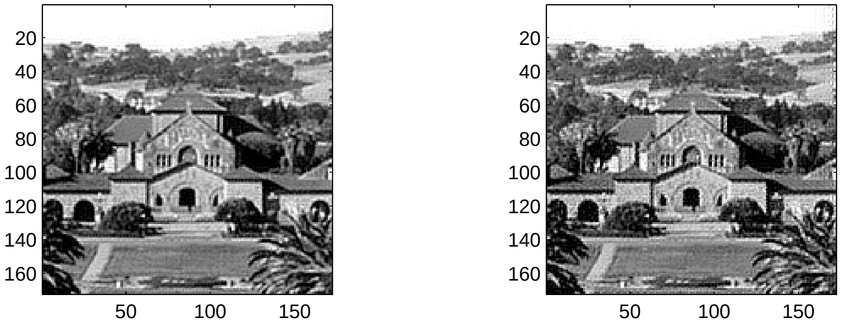


Figure 2.9: Original Stanford image.

grainier, with sawtooth artifacts along straight lines. Figures 2.11 and 2.12 show results with 16 and 20 data frames, respectively. We note that for the case with 16 frames, reconstruction quality does not degrade appreciably until registration error level at 0.25 pixels compared to 0.15 pixels with 10 frames. For the overdetermined data experiment with 20 frames, qualitative degradations occur at an even lower level of registration accuracy, at 0.30 pixels misregistration. We reach the same conclusion as we did examining the MSE graph. With more data available, the superresolution is more resistant to registration errors. These figures suggest a direct linear relationship between data availability and robustness to misregistration.

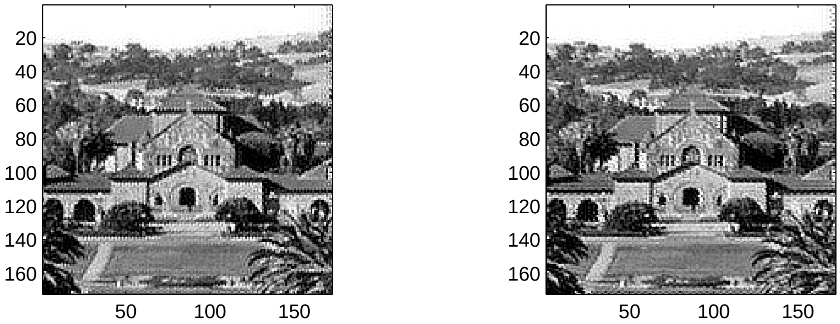
2.8 Conclusion

Image registration is an important aspect of superresolution. In this chapter, we introduced a projection-based framework for estimating frame motion. We derived an expression for affine motion in the projection space. For purely translational 2-D motion, this expression allowed us to reduce the problem to two independent 1-D translational motion problems. Registration timing results comparing the projected and original versions of two popular motion estimation techniques illustrated the benefits of this reduction in complexity. Accuracy results demonstrated that there is no significant loss in performance when the problem is solved using projection. We also presented a theoretical bound on performance loss of projected methods compared to the original versions. In the second part of the chapter, we showed



Reconstruction without registration errors

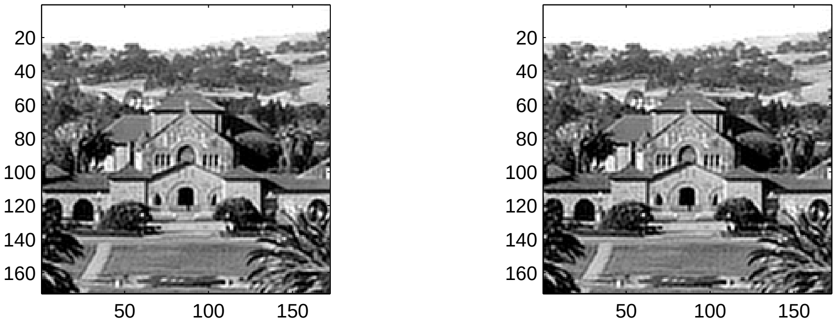
Reconstruction with 0.05 pixels misregistration



Reconstruction with 0.1 pixels misregistration

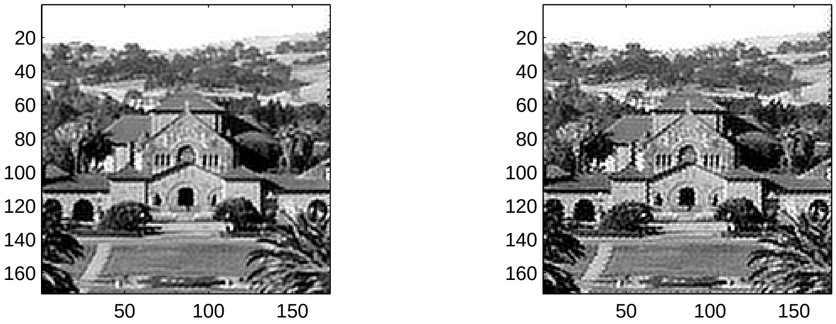
Reconstruction with 0.15 pixels misregistration

Figure 2.10: Superresolution result with misregistration: underdetermined.



Reconstruction without registration errors

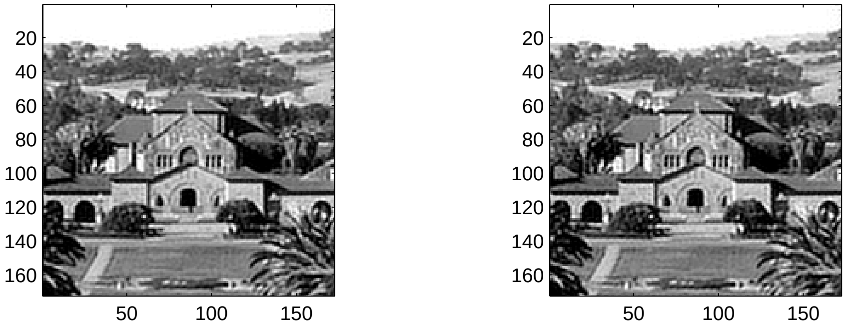
Reconstruction with 0.05 pixels misregistration



Reconstruction with 0.15 pixels misregistration

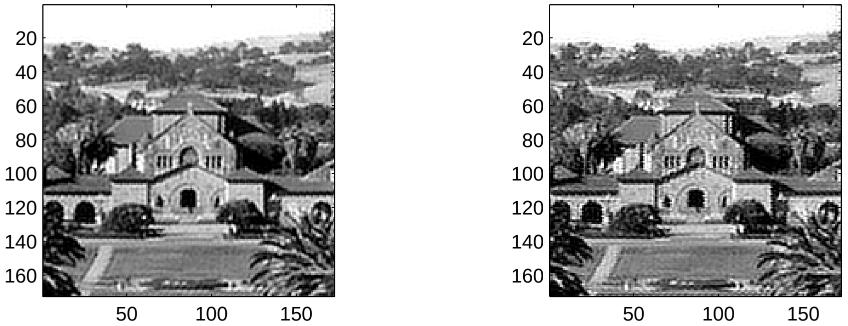
Reconstruction with 0.25 pixels misregistration

Figure 2.11: Superresolution result with misregistration: square.



Reconstruction without registration errors

Reconstruction with 0.1 pixels misregistration



Reconstruction with 0.2 pixels misregistration

Reconstruction with 0.3 pixels misregistration

Figure 2.12: Superresolution result with misregistration: overdetermined.

quantitative and qualitative effects of misregistration on superresolution results. Our experiments illustrated that as expected, reconstructed image quality degrades as registration error increases. Furthermore, given additional data, superresolution becomes more robust to misregistration.

Chapter 3

Preconditioning for Superresolution

3.1 Introduction

Superresolution is a computationally intensive problem, typically involving tens of thousands of unknowns and data values. For example, superresolving a sequence of 50×50 pixel LR frames by a factor of 4 in each spatial dimension involves $200 \times 200 = 40000$ unknown pixel values in the HR image. With 10 data frames, the number of data samples is $10 \cdot (50 \times 50) = 25000$. A practical superresolution algorithm must be efficient in data floating point calculations. Storage and data I/O are also a significant concern. We recall the model equation for superresolution,

$$\begin{aligned} \mathbf{f}_k &= DC_kE_k\mathbf{x} + \mathbf{n}_k, \quad 1 \leq k \leq p, && (3.1) \\ &= H_k\mathbf{x} + \mathbf{n}_k, && (3.2) \end{aligned}$$

where p is the number of available frames, $\mathbf{f}_k$ is an $N \times 1$ vector representing the kth $m \times n$ ($N = mn$ pixels) LR image in columnwise order. If l is the resolution enhancement factor in each direction, $\mathbf{x}$ is an $l^2N \times 1$ vector representing the $lm \times ln$ HR image in columnwise order, E_k is an $l^2N \times l^2N$ warping matrix that represents the relative motion between frame k and a reference frame, C_k is a blur matrix of size $l^2N \times l^2N$, D is the $N \times l^2N$ uniform down-sampling matrix, and $\mathbf{n}_k$ is the $N \times 1$ vector representing additive noise. Stacking these equations on top of one another, we get the entire system:

$$\begin{aligned} \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_p \end{bmatrix} &= \begin{bmatrix} H_1 \\ \vdots \\ H_p \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_p \end{bmatrix} && (3.3) \\ \mathbf{f} &= H\mathbf{x} + \mathbf{n}. && (3.4) \end{aligned}$$

In the example above, $N = 2500$, $l = 4$, and $p = 10$. Hence, the dimensions of the operator H are 25000×40000. Storing H even in sparse format is cumbersome. A more elegant approach would treat H as a sequence of operators. Multiplication by H is done by sequentially applying these operators. Thus, we would not need to store H explicitly. Since we have assumed small and global frame to frame motions and blur with local support, the operators E_k's and C_k's can be applied locally at each image pixel. Matrix-vector operations involving the operator H can be done very efficiently through this process. Previous work on superresolution has not put adequate emphasis on this and other computational aspects of the problem. Many iterative approaches have been proposed. Patti *et al.* [69] and Stark and Oskoui [81] proposed projection onto convex sets (POCS) algorithms, which defined sets of closed convex constraints whose intersection contains the HR estimate, and successively projected an arbitrary initial estimate onto these constraint sets. Others [46, 59, 9] adopted a related method, the iterative back-projection method, frequently used in computer aided tomography. Cheeseman *et al.* [19] used the standard Jacobi's method, and Hardie *et al.* [39] proposed a steepest descent algorithm in combination with block matching to compute simultaneously the HR image and the registration parameters. Although they are usually robust to noise and allow some modeling flexibility, projection-based algorithms are also known for their low rate of convergence. More recently, Hardie *et al.* [39], Connolly and Lane [20] and Chan *et al.* [16] have considered CG methods for Tikhonov regularized superresolution. However, Hardie *et al.* [39] and Connolly and Lane [20] did not consider preconditioning for their CG algorithm, and Chan *et al.*'s [16] preconditioner is only applicable for multisensor arrays. In this chapter, we present efficient block preconditioners that take advantage of the inherent structures in the superresolution system matrix H to accelerate convergence for Krylov subspace methods. The preconditioners introduced in this chapter are frame-based, consisting of components approximating the degradation processes for the LR frames. We assume that the frame-to-frame global motions are, or can be approximated to be, purely translational. We also assume that the blurring process for each LR frame is spatially invariant, although not necessarily temporally invariant, i.e., not the same from one frame to the next.

3.2 Regularization

This section briefly introduces regularization techniques for least squares problems, a topic to be discussed in greater detail in Chapter 4. For simplicity, we consider only least squares formulation for square/overdetermined systems here. This is necessary in order to present the preconditioners in the proper context. We will present a more general regularized least squares framework covering all underdetermined, square, and overdetermined systems in Chapter 4.

The operators C_k, related to the point spread functions (PSF), are derived from the discretization of a compact operator, so H is ill-conditioned [4]. Thus, even small changes in $\mathbf{f}$ can result in wild oscillations in approximations to $\mathbf{x}$ when (3.4) is solved directly. To obtain a reasonable estimate for $\mathbf{x}$ we reformulate the problem as a regularized unconstrained minimization problem,

$$\min_{\mathbf{x}} \;\; \|H\mathbf{x} - \mathbf{f}\|_2^2 + \lambda\|L\mathbf{x}\|_2^2, \tag{3.5}$$

where L is a regularization matrix of full rank, and λ is related to the Lagrange multiplier. Equivalently, we can write (3.5) as

$$\min_{\mathbf{x}} \left\| \begin{pmatrix} H \\ \sqrt{\lambda}L \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \right\|_2^2 . \tag{3.6}$$

In this formulation, L serves as a stabilization matrix, and the new system is better conditioned. While a simple and effective regularization matrix can be the identity I, L can also incorporate some prior knowledge of the problem, e.g. degree of smoothness [37]. In Chapter 4, we show the reduction of (3.5) to standard form, i.e. $L = I$. For the rest of this chapter, we assume that L is the identity. The solution to the regularized least squares problem (3.5) with $L = I$ is

$$\mathbf{x} = (H^T H + \lambda I)^{-1} H^T \mathbf{f}. \tag{3.7}$$

In the above formulation, λ is the regularization parameter. A larger λ corresponds to a better conditioned system, but the new system is also farther away from the original system we wish to solve. We adopt the generalized cross-validation (GCV) criterion (cf. [30]) for calculating the optimum regularization parameter. One reason why GCV is so widely used is that it does not require any a priori knowledge. The criterion estimates the parameter directly from raw data. GCV is also known to be relatively insensitive to large individual equation errors, i.e. outliers. For overdetermined systems, it has been shown [30] that the asymptotically optimum regularization parameter according to GCV is given by:

$$\lambda_{GCV} = \text{argmin}_{\lambda} \frac{\|(HH^T + \lambda I)^{-1}\mathbf{f}\|_2}{tr((HH^T + \lambda I)^{-1})}. \tag{3.8}$$

In Chapter 4, we derive this same expression for underdetermined systems and describe an inexpensive way to approximate the objective function in (3.8) using preconditioners proposed in this chapter. Cross-validation and GCV are also described in further detail. A more accurate and robust method to estimate (3.8) using Gauss-type quadrature rules is described in Chapter 5.

3.3 Preconditioning for Conjugate Gradient

As we described earlier, superresolution is computationally intensive. The number of unknowns, the same as the number of pixels in the HR image, is typically in the tens or hundreds of thousands. The computational advantages of CG and Krylov subspace methods over stationary iterative methods, e.g. Jacobi, Gauss-Seidel, SOR, Kaczmarz, etc., for solving large, sparse, symmetric positive definite linear systems are well-known. Krylov subspace methods converge at a greater rate than stationary iterative methods especially with proper preconditioner acceleration. Furthermore, unlike some stationary methods, Krylov subspace methods do not have to tune free parameters. The main computational burden of Krylov subspace methods is the matrix-vector products with the system matrix. For our superresolution problem, these products may be performed efficiently as we described in the introduction to this chapter. For completeness, we include the standard CG algorithm for symmetric

```
$\mathbf{r}_0 = A\mathbf{x}_0 - \mathbf{b}$;
$\mathbf{p}_0 = \mathbf{r}_0$;
$\gamma_0 = (\mathbf{r}_0, \mathbf{r}_0)$;

for $i = 0 : max\_it$
    $\mathbf{q}_i = A\mathbf{p}_i$;
    $\alpha_i = \gamma_i/(\mathbf{q}_i, \mathbf{p}_i)$;
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha\mathbf{p}_i$;
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha\mathbf{q}_i$;
    $\gamma_{i+1} = (\mathbf{r}_{i+1}, \mathbf{r}_{i+1})$;
    if $\gamma_{i+1} < tol * \|\mathbf{b}\|_2^2$
        break;
    end
    $\beta_i = \gamma_{i+1}/\gamma_i$;
    $\mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_i\mathbf{p}_i$;
end
```

Table 3.1: Conjugate gradient algorithm.

positive definite systems $A\mathbf{x} = \mathbf{b}$ in Table 3.1.

The convergence rate for CG [89] is dependent on the distribution of eigenvalues of the system matrix. The method works well on matrices that are either well-conditioned or have just a few distinct eigenvalues; see [33, p. 525]. Preconditioning is a technique used to transform the original system into one with the same solution, but which can be solved by the iterative solver more efficiently [72, p. 245]. For a symmetric positive definite matrix A, the preconditioner M can be thought of as a symmetric positive definite matrix that approximates A in some sense. Then instead of solving the original system

$$A\mathbf{x} = \mathbf{b}, \tag{3.9}$$

we work with the equivalent

$$M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}. \tag{3.10}$$

```
$\mathbf{r}_0 = \mathbf{x}_0 - \mathbf{b}$;
$\mathbf{z}_0 = M^{-1}\mathbf{r}_0$;
$\mathbf{p}_0 = \mathbf{z}_0$;
$\gamma_0 = (\mathbf{r}_0, \mathbf{z}_0)$;

for $i = 0 : max\_it$
    $\mathbf{q}_i = A\mathbf{p}_i$;
    $\alpha_i = \gamma_i / (\mathbf{q}_i, \mathbf{p}_i)$;
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha\mathbf{p}_i$;
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha\mathbf{q}_i$;
    if $\|\mathbf{r}_{i+1}\|_2 < tol * \|\mathbf{b}\|_2$
        break;
    end
    $\mathbf{z}_{i+1} = M^{-1}\mathbf{r}_{i+1}$;
    $\gamma_{i+1} = (\mathbf{r}_{i+1}, \mathbf{z}_{i+1})$;
    $\beta_i = \gamma_{i+1} / \gamma_i$;
    $\mathbf{p}_{i+1} = \mathbf{z}_{i+1} + \beta_i\mathbf{p}_i$;
end
```

Table 3.2: Preconditioned conjugate gradient algorithm.

The new preconditioned system $M^{-1}A$ is no longer symmetric positive definite, so the usual Euclidean inner product $(\cdot, \cdot)$ in the algorithm must be replaced by the M-inner product. Table 3.2 contains the modified preconditioned CG (PCG) algorithm. Note that with $M = I$, the identity matrix, the PCG algorithm is equivalent to the unpreconditioned one.

In general, a preconditioner M for a matrix A should satisfy the following criteria [5, p. 253]:

- The cost of computing M should be low.
- The computational cost of solving a linear system with matrix M should be low.
- The iterative solver should converge much faster with $M^{-1}A$ than with A.

For image superresolution, we would like to precondition the inverse step,

$$(H^T H + \lambda I)\mathbf{y} = H^T \mathbf{f}. \tag{3.11}$$

From our spatial invariance assumption of the PSF, each submatrix H_i of H (in (3.2) corresponding to a frame i has block Toeplitz-like structure. We propose preconditioners derived from circulant approximations to H_i. We demonstrate that our preconditioners satisfy the three criteria described above. First of all, circulant matrices have the useful property that they can be diagonalized by the discrete Fourier transform (cf. [17]). The eigendecomposition of a circulant matrix C can be written as follows:

$$C = F^* \Lambda F, \tag{3.12}$$

where F is the unitary discrete Fourier transform matrix and Λ is a diagonal matrix containing the eigenvalues of C. We can also compute the eigenvalues of C by taking the FFT of its first column. Using this special property, we do not need to construct our preconditioners explicitly. We only need to store the entries of their first columns. Therefore, the cost of constructing the preconditioners is negligible. Additionally, operations involving circulant matrices can be done efficiently by FFTs. The cost of solving a linear system with a circulant coefficient matrix is 2 FFTs. For a block matrix with circulant blocks such as our preconditioners, we need to solve a linear system with a block coefficient matrix with diagonal blocks in addition to the 2 FFTs. To study the convergence behavior of the PCG procedure described here, we will prove bounds on the maximum number of preconditioned iterations required for exact convergence.

We first examine the structure of the system matrix arising from superresolution. Submatrix H_i, the block row of H associated with frame i, has the form:

$$H_i = \begin{bmatrix} T_{i1} & T_{i2} & \cdots & T_{il^2} \end{bmatrix}, \tag{3.13}$$

where each block T_{ij} is an $N \times N$ 'nearly' Toeplitz upper band matrix[1]. In fact, T_{ij} only has nonzero entries on a few superdiagonals close to the main diagonal because the support of the PSF is local. For example, in the simple case of superresolving a sequence of four, 2×4 pixel, LR frames by a factor of 2 in each dimension, T_{23} has the following structure:

$$T_{23} = \begin{bmatrix} 0 & t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.14}$$

The zero entry in the middle of the nonzero diagonal is the result of Dirichlet boundary condition imposed on the LR frames. We build our preconditioners for (3.11) with two approximation steps. We first approximate H_i by a block matrix $\tilde{H}_i = (\tilde{T}_{ij})$, whose blocks $\tilde{T}_{ij}$ are Toeplitz. We construct $\tilde{T}_{ij}$ from T_{ij} by filling in the zero entries along nonzero diagonals, so that $\tilde{T}_{ij}$ is just a low rank change from T_{ij}. For example, the approximation to T_{23} would be:

$$\tilde{T}_{23} = \begin{bmatrix} 0 & t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.15}$$

---

[1] Almost all entries along the diagonals of T_{ij} are constant.

We can think of $\tilde{H} = (\tilde{H}_i)$ as a small rank change from H. We quantify this approximation more concretely in the next section.

The second step approximates the block matrix $\tilde{H}$ (with Toeplitz blocks $\tilde{H}_i$), with circulant type preconditioners. The following section outlines two preconditioners for upper band Toeplitz matrices. We first describe these preconditioners for point Toeplitz matrices. Extensions of these preconditioners and their convergence properties to the block case and application to regularized least squares problem (3.11) are then examined.

3.3.1 Circulant Preconditioner

The first preconditioner, originally developed by Strang [17], completes a Toeplitz matrix T by copying the central diagonals. For an upper triangular banded Toeplitz matrix T with bandwidth b,

$$T = \begin{bmatrix} t_0 & \cdots & t_b & & \\ & \ddots & & \ddots & \\ & & t_0 & \cdots & t_b \\ & & & \ddots & \vdots \\ & & & & t_0 \end{bmatrix}, \tag{3.16}$$

the preconditioner C_S is simply

$$C_S = \begin{bmatrix} t_0 & \cdots & \cdots & t_b & & \\ & \ddots & & & \ddots & \\ & & t_0 & \cdots & \cdots & t_b \\ t_b & & & t_0 & \cdots & t_{b-1} \\ \vdots & \ddots & & & \ddots & \vdots \\ t_1 & \cdots & t_b & & & t_0 \end{bmatrix}. \tag{3.17}$$

The preconditioner C_S is only a rank b change from T because only the last b rows (the first b columns) of C_S are different from T. The following lemma quantifies the quality of C_S as an approximation to T.

Lemma 1 *Let T be an upper band Toeplitz matrix with bandwidth less than or equal to b. Let C_S be the circulant preconditioner to T as described above. If C_S is nonsingular, then at most b eigenvalues of the preconditioned system $C_S^{-1}T$ are not equal to 1.*

Proof. From (3.16) and (3.17), we have

$$T = C_S - K, \text{rank}(K) \leq b. \tag{3.18}$$

The matrix K represents the wraparound entries in the bottom left of C_S. Multiplying both sides by C_S^{-1},

$$C_S^{-1}T = I - C_S^{-1}K, \tag{3.19}$$

with $\text{rank}(C_S^{-1}K) \leq b$. Therefore, the preconditioned system contains at most b eigenvalues not equaling to 1.

■

For a block matrix $T = (T_{ij})$ with Toeplitz blocks T_{ij} with bandwidth b_{ij}, the block version of the preconditioner is $C_S = (C_{ij})$, where each block C_{ij} is Strang's circulant approximation to T_{ij}. We can extend the result of Lemma 1 to the block case. We have deliberately left the range of indices i, j unspecified because this construction technique and result is applicable for rectangular block matrices as well.

Lemma 2 *Let T be a matrix consisting of Toeplitz blocks T_{ij} and C_S be the preconditioner with circulant blocks C_{ij}. Suppose that the blocks T_{ij} are upper banded with bandwidths b_{ij}. For any row block index I and column block index J, let*

$$b_{I,\max} = \max_j b_{Ij}, \quad b_{\max,J} = \max_i b_{iJ}. \tag{3.20}$$

Then

$$C_S = T - K, \text{ where } \operatorname{rank}(K) \leq \min\left\{\sum_J b_{\max,J}, \sum_I b_{I,\max}\right\}. \tag{3.21}$$

Proof. The lemma is proved by counting the number of rows and columns in the preconditioner C_S that are different in T. For row block index I, let

$$T(I,:) = \begin{bmatrix} T_{I1} & T_{I2} & \cdots \end{bmatrix} \tag{3.22}$$

$$C(I,:) = \begin{bmatrix} C_{I1} & C_{I2} & \cdots \end{bmatrix}. \tag{3.23}$$

For each block T_{Ij}, the corresponding preconditioner block C_{Ij} is different from T_{Ij} in the last b_{Ij} rows. Thus, for the whole row block, the last $b_{I,\max} = \max_j b_{Ij}$ rows of $C(I,:)$ are different from $T(I,:)$. Summing the number of changed rows for all row blocks, we get $\sum_I b_{I,\max}$ as the total number of changed rows from T to C_S. A similar counting argument gives $\sum_J b_{\max,J}$ as the total number of changed columns. Therefore,

$$C_S = T - K, \tag{3.24}$$

where K has $\sum_I b_{I,\max}$ nonzero rows and $\sum_J b_{\max,J}$ nonzero columns. This implies that

$$\operatorname{rank}(K) \leq \min\left\{\sum_J b_{\max,J}, \sum_I b_{I,\max}\right\}. \tag{3.25}$$

■

We apply this preconditioning technique to solve the regularized least squares problem (3.11). Recall that we approximated H with a block-Toeplitz matrix

$$\tilde{H} = \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} & \cdots & \tilde{T}_{1l^2} \\ \tilde{T}_{21} & \tilde{T}_{22} & \cdots & \tilde{T}_{2l^2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{T}_{p1} & \tilde{T}_{p2} & \cdots & \tilde{T}_{pl^2} \end{bmatrix} \tag{3.26}$$

with each block $\tilde{T}_{ij}$ being Toeplitz, l the resolution enhancement factor, and p the number of LR frames. The block-circulant preconditioner proposed is a block matrix with $N \times N$ circulant blocks

$$C_S = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1l^2} \\ C_{21} & C_{22} & \cdots & C_{2l^2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{p1} & C_{p2} & \cdots & C_{pl^2} \end{bmatrix} \tag{3.27}$$

such that C_{ij} is the circulant approximation to $\tilde{T}_{ij}$. Note that we can diagonalize $C_{ij} = F_N^* \Lambda_{ij} F_N$ by the Fourier transform, where F_N is the $N \times N$ discrete Fourier transform matrix, and Λ_{ij} is a diagonal matrix of eigenvalues of C_{ij}. Therefore,

$$C_S = \mathcal{F}_p^* \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \cdots & \Lambda_{1l^2} \\ \Lambda_{21} & \Lambda_{22} & \cdots & \Lambda_{2l^2} \\ \vdots & \vdots & \ddots & \vdots \\ \Lambda_{p1} & \Lambda_{p2} & \cdots & \Lambda_{pl^2} \end{bmatrix} \mathcal{F}_{l^2}, \tag{3.28}$$

with $\mathcal{F}_k = diag(\underbrace{F_N, F_N, \ldots, F_N}_{k})$. The preconditioner for least squares equation (3.11) is

$$C_S^T C_S + \lambda I \quad = \quad \mathcal{F}_{l^2}^* (\Lambda^* \Lambda + \lambda I) \mathcal{F}_{l^2}, \tag{3.29}$$

where $\Lambda = (\Lambda_{ij})$ is a block matrix with diagonal blocks of eigenvalues. The decomposition above illustrates how our block-circulant preconditioner may be constructed and applied efficiently with FFTs. Using Lemma 2, we now show the preconditioner's effectiveness.

Theorem 1 *Let T be a block-Toeplitz matrix with Toeplitz blocks T_{ij} and bandwidths b_{ij}, C_S be the block-circulant preconditioner for T as described above, and*

$$r = \min\left\{\sum_J b_{\max,J}, \sum_I b_{I,\max}\right\}. \tag{3.30}$$

For $\lambda > 0$, we have the following result:

$$(C_S^T C_S + \lambda I)^{-1}(T^T T + \lambda I) = I - \mathcal{K}, \tag{3.31}$$

$$\operatorname{rank}(\mathcal{K}) \le 3r. \tag{3.32}$$

Proof. Since C_S is the block-circulant approximation to T, Lemma 2 gives us

$$C_S = T - K, \quad \operatorname{rank}(K) \le r. \tag{3.33}$$

Therefore,

$$\begin{aligned}(C_S^T C_S + \lambda I)^{-1}(T^T T + \lambda I) &= (C_S^T C_S + \lambda I)^{-1} \times \\ &\quad (C_S^T C_S + K^T C_S + C_S^T K + K^T K + \lambda I) \\ &= I - (C_S^T C_S + \lambda I)^{-1}(K^T C_S + C_S^T K + K^T K) \\ &= I - \mathcal{K}. \end{aligned} \tag{3.34}$$

Because $\operatorname{rank}(K) \le r$,

$$\operatorname{rank}(K^T C_S + C_S^T K + K^T K) \le 3r. \tag{3.35}$$

■

The implication of this theorem is that with a circulant-block preconditioner, at most $3r+1$ PCG iterations are required to solve an upper banded Toeplitz-block linear system exactly. The scalar r depends directly on the bandwidths of the blocks and the number of blocks. Note that the theorem may also be used for bounding the number of PCG iterations to solve for an upper banded point Toeplitz system by setting the number of blocks to be 1 and r to be the bandwidth of the system.

For the case $L \neq I$, we can precondition in a similar manner if L can be approximated well by some block-circulant matrix C_L that has the same block structure as C. That is, if

$$C_L = \mathcal{F}_{l^2}^* \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \cdots & \Gamma_{1l^2} \\ \Gamma_{21} & \Gamma_{22} & \cdots & \Gamma_{2l^2} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{l^21} & \Gamma_{l^22} & \cdots & \Gamma_{l^2l^2} \end{bmatrix} \mathcal{F}_{l^2}. \tag{3.36}$$

Hence, we can precondition the matrix $H^TH + \lambda L^TL$ by

$$C_S^TC_S + \lambda C_L^TC_L \;=\; \mathcal{F}_{l^2}^*(\Lambda^*\Lambda + \lambda\Gamma^*\Gamma)\mathcal{F}_{l^2}, \tag{3.37}$$

where $\Gamma = (\Gamma_{ij})$ is also a block matrix with diagonal blocks. In practice, aside from the identity, operators derived from discrete approximation to the kth derivative are also popular choices for L. With a proper arrangement of the rows, these operators also have banded block-Toeplitz structure, and we can extend the techniques of preconditioning as in (3.37).

3.3.2 Approximate Inverse Preconditioner

Our second preconditioner, originally developed by Hanke and Nagy [38], is an approximate inverse preconditioner for an upper banded Toeplitz matrix $T_{N\times N}$ with bandwidth less than or equal to b. It is constructed as follows. First we embed T

within an $(N+b)\times(N+b)$ circulant matrix C_{HN} of the form:

$$C_{HN} = \begin{bmatrix} T & U \\ V & W \end{bmatrix}, \tag{3.38}$$

where

$$W = \begin{bmatrix} t_0 & t_1 & \cdots & t_{b-1} \\ & t_0 & \cdots & t_{b-2} \\ & & \ddots & \vdots \\ & & & t_0 \end{bmatrix}, \tag{3.39}$$

$$U = \begin{bmatrix} 0 \\ L \end{bmatrix}, V = \begin{bmatrix} L & 0 \end{bmatrix}, L = \begin{bmatrix} t_b & & \\ \vdots & \ddots & \\ t_1 & \cdots & t_b \end{bmatrix}. \tag{3.40}$$

Figure 3.1 illustrates this construction by showing the sparsity pattern of C_{HN}. If C_{HN} is nonsingular, we partition C_{HN}^{-1} as

$$C_{HN}^{-1} = \begin{bmatrix} M & \tilde{U} \\ \tilde{V} & \tilde{W} \end{bmatrix}, \tag{3.41}$$

where M is the $N\times N$ leading principal submatrix. The matrix M is the approximate inverse preconditioner for T. We have similar results as Lemmas 1 and 2 for the approximate inverse preconditioner.

Lemma 3 *Let T be an upper band Toeplitz matrix with bandwidth less than or equal to b. Let C_{HN} be the circulant extension matrix as described above. If C_{HN} is nonsingular and M is the leading $N\times N$ principal matrix of C_{HN}^{-1}, then the preconditioned system MT has at most b eigenvalues not equal to 1.*

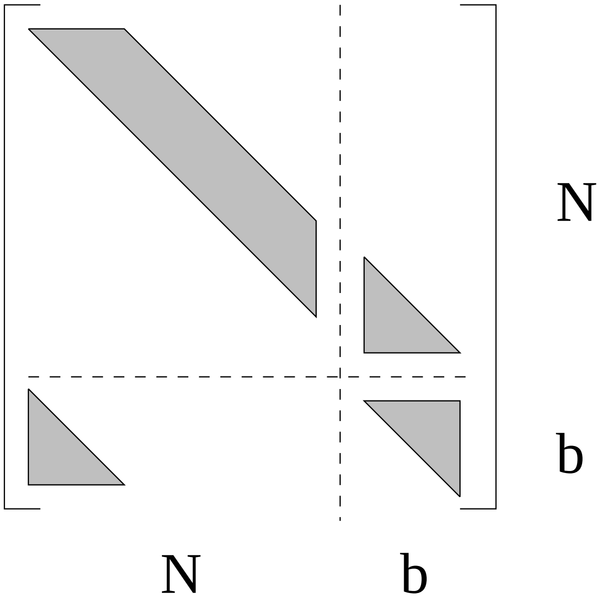


Figure 3.1: Sparsity pattern of the circulant extension matrix.

Proof. We start with the identity

$$
\begin{aligned}
I &= C_{HN}C_{HN}^{-1} && (3.42)\\
&= \begin{bmatrix} M & \tilde{U} \\ \tilde{V} & \tilde{W} \end{bmatrix}\begin{bmatrix} T & U \\ V & W \end{bmatrix} && (3.43)\\
&= \begin{bmatrix} MT+\tilde{U}V & MU+\tilde{U}W \\ \tilde{V}T+\tilde{W}V & \tilde{V}U+\tilde{W}U \end{bmatrix}. && (3.44)
\end{aligned}
$$

Therefore,

$$MT = I - \tilde{U}V. \tag{3.45}$$

The dimensions of $\tilde{U}$ and V are $N \times b$ and $b \times N$ respectively, so $\operatorname{rank}(\tilde{U}V) \le b$. The preconditioned system MT is at most b eigenvalues different from the identity matrix.

■

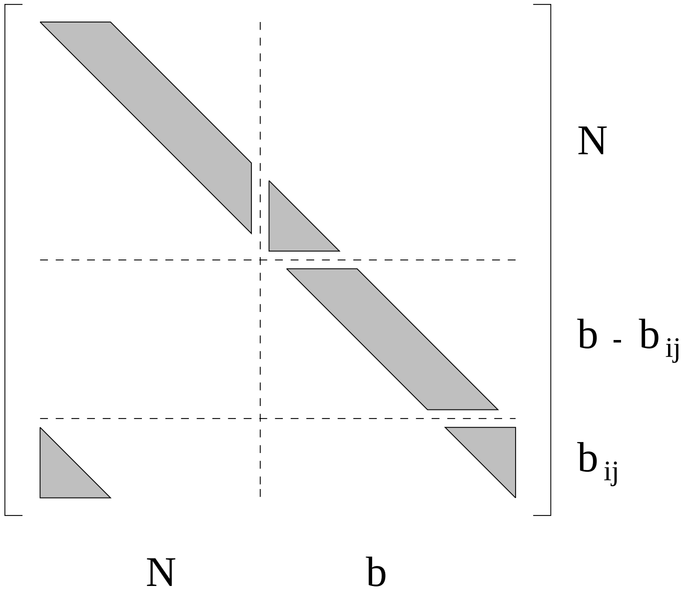


Figure 3.2: Sparsity pattern of the block circulant extension matrix.

The construction of a preconditioner for a block matrix T with Toeplitz blocks T_{ij} is slightly more complicated. Let $b = \max_{i,j} b_{ij}$ be the largest bandwidth among the blocks. We embed each $N \times N$ block T_{ij} in a circulant $(N+b) \times (N+b)$ extension C_{ij}. Unlike the point matrix case, each block T_{ij} may have to be extended by more than its bandwidth ($b > b_{ij}$). Figure 3.2 illustrates how this is done. The circulant extension matrix C_{ij} may be decomposed as

$$C_{ij} = \begin{bmatrix} T_{ij} & U_{ij} \\ V_{ij} & W_{ij} \end{bmatrix}, \tag{3.46}$$

where U_{ij}, V_{ij} are $N \times b$, $b \times N$ matrices respectively with rank b_{ij}, and W_{ij} is a $b \times b$ matrix with rank b. For $T = (T_{ij})_{1 \le i \le p, 1 \le j \le q}$, the complete block circulant extension matrix has the following form:

$$C_{HN} = \begin{bmatrix} T_{11} & U_{11} & \cdots & \cdots & T_{1q} & U_{1q} \\ V_{11} & W_{11} & \cdots & \cdots & V_{1q} & W_{1q} \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ T_{p1} & U_{p1} & \cdots & \cdots & T_{pq} & U_{pq} \\ V_{p1} & W_{p1} & \cdots & \cdots & V_{pq} & W_{pq} \end{bmatrix}. \tag{3.47}$$

Now let P be the permutation matrix that groups blocks with the same lettering together:

$$PC_{HN}P^T = \begin{bmatrix} T & U \\ V & W \end{bmatrix}, \tag{3.48}$$

$$T = (T_{ij})_{1\le i\le p, 1\le j\le q}, \quad U = (U_{ij})_{1\le i\le p, 1\le j\le q}, \tag{3.49}$$

$$V = (V_{ij})_{1\le i\le p, 1\le j\le q}, \quad W = (W_{ij})_{1\le i\le p, 1\le j\le q}. \tag{3.50}$$

The normal equation matrix for this extended block-circulant matrix is

$$C_{HN}^T C_{HN} + \lambda I = P^T \begin{bmatrix} T^T & V^T \\ U^T & W^T \end{bmatrix} \begin{bmatrix} T & U \\ V & W \end{bmatrix} P + \lambda I \tag{3.51}$$

$$= P^T \begin{bmatrix} T^T T + U^T U + \lambda I & T^T V + U^T W \\ V^T T + W^T U & V^T V + W^T W + \lambda I \end{bmatrix} P. \tag{3.52}$$

Now decompose the inverse of the 2 × 2 block matrix into blocks of the same size:

$$\begin{bmatrix} T^T T + U^T U + \lambda I & T^T V + U^T W \\ V^T T + W^T U & V^T V + W^T W + \lambda I \end{bmatrix}^{-1} = \begin{bmatrix} M & X \\ Y & Z \end{bmatrix}. \tag{3.53}$$

The matrix M is our block approximate inverse preconditioner for $T^T T + \lambda I$. We have the following theorem, which bounds the number of eigenvalues of the preconditioned system.

Theorem 2 *Let T be a block-Toeplitz matrix with Toeplitz blocks T_{ij} and bandwidths b_{ij}. Let M be the block approximate inverse preconditioner as constructed above. For row block index I and column block index J, let*

$$b_{I,\max} = \max_j b_{Ij}, \quad b_{\max,J} = \max_i b_{iJ} \tag{3.54}$$

and

$$r = \min \left\{ \sum_J b_{\max,J}, \sum_I b_{I,\max} \right\}. \tag{3.55}$$

Then

$$M(T^T T + \lambda I) = I - \mathcal{K}, \tag{3.56}$$

$$\operatorname{rank}(\mathcal{K}) \leq 3r. \tag{3.57}$$

Proof. From (3.53), we have

$$M(T^T T + U^T U + \lambda I) + X(V^T T + W^T U) = I. \tag{3.58}$$

Hence,

$$M(T^T T + \lambda I) = I - (MU^T U + XV^T T + XW^T U). \tag{3.59}$$

The block matrix U consists of pq blocks U_{ij}. Each block U_{ij} consists mostly of zero columns except for the last b_{ij} rows and the first b_{ij} columns. In fact, for a specific block row index I, the number of nonzero columns of U_{ij} is $b_{I,\max}$ and for a specific block column index J, the number of nonzero rows is $b_{\max,J}$. Hence, the total number of nonzero columns is $\sum_I b_{I,\max}$, and the total number of nonzero rows is $\sum_J b_{\max,J}$. We can conclude that

$$\operatorname{rank}(U) \leq r. \tag{3.60}$$

A similar argument could be made for V. Therefore, (3.56) and (3.57) hold.

Theorem 2 gives us the same upper bound on the number of non-unit eigenvalues as that of Theorem 1. We expect the performance of the circulant-block preconditioner and the block approximate inverse preconditioner to be comparable when applied to our Toeplitz-block system $\tilde{H}$.

3.4 Convergence Bounds for Superresolution

The previous section describes construction techniques and provides some eigenvalue bounds for our preconditioners. This section assembles those results to bound the number of PCG iterations necessary for exact convergence for superresolution. Recall that we build our preconditioners using two approximation steps. The first step approximates the superresolution system matrix with a block matrix with Toeplitz blocks. The following theorem bounds the rank of this update.

Theorem 3 *Let H be the block system matrix representing the forward model for superresolution and $\tilde{H}$ be the Toeplitz-block matrix approximating H as described in the previous section. Then*

$$\tilde{H} = H - K, \ \ \operatorname{rank}(K) \leq p(m+n-1), \tag{3.61}$$

where m, n are the dimensions of an LR frame, and p is the number of available frames.

Proof. Figure 3.3 gives a conceptual look at the forward model for superresolution. Each LR pixel value is a weighted average of a box of HR pixel intensities. The boundary LR pixels cover areas outside the HR pixel grid. The rows in the system matrix H associated with these boundary pixels have zero entries corresponding to

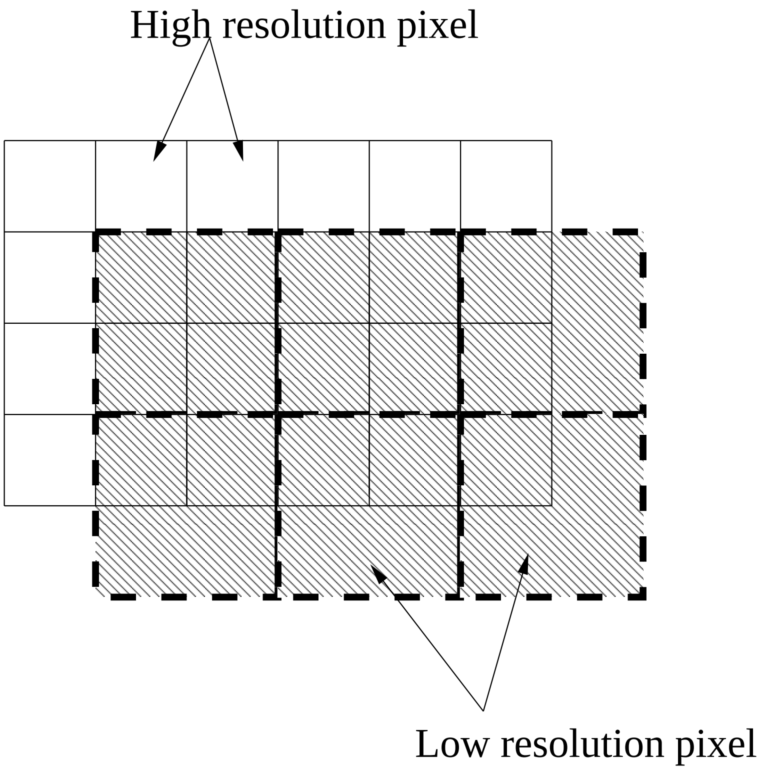


Figure 3.3: LR pixel values as a weighted average of HR pixel values.

these areas because of Dirichlet boundary conditions. These are the zero gaps in the nearly Toeplitz diagonals of H. The updated Toeplitz-block matrix $\tilde{H}$ is just H with these gaps filled with appropriate entries. For each frame, there are $m+n-1$ boundary pixels. Thus, the number of rows that are updated from H to $\tilde{H}$ is at most $p(m+n-1)$.

■

The second step approximates the Toeplitz-block matrix with a circulant-block matrix. Theorems 1 and 2 bound the rank of this approximation step for both preconditioners by $3r$, where

$$r = \min\left\{\sum_J b_{\max,J}, \sum_I b_{I,\max}\right\}, \tag{3.62}$$

and $b_{I,\max}$ and $b_{\max,J}$ are maximum bandwidth of row I and column J, respectively. For superresolution, the following theorem bounds the bandwidth for each block.

Theorem 4 *The superresolution system matrix H in (3.13) and its block Toeplitz approximate $\tilde{H}$ have blocks with bandwidths bounded by* $\min(m,n)+1$*, where m and n are the height and width of an LR frame, respectively.*

Combining results from Theorems 1–4, we obtain a convergence bound for PCG for superresolution.

Theorem 5 *Let H be the superresolution system matrix for p LR frames each $m \times n$ pixels in dimensions, with l the desired enhancement factor in each dimension. PCG based on block versions of either Strang's or Hanke-Nagy's preconditioner achieves exact solution in at most* $p(m+n-1)+3\min(p,l^2)(\min(m,n)+1)+1$ *iterations.*

Proof. Using either preconditioner, from Theorems 1 and 2 the second step approximation update rank is bounded by $3r$, with r being defined as in (3.62). The matrix H is a block matrix with $p \times l^2$ blocks each $mn \times mn$ in dimension. From Theorem 3, we get

$$r \leq \min(p,l^2)(\min(m,n)+1). \tag{3.63}$$

Therefore, the total rank of the difference between the preconditioners and the original system matrix is $p(m+n-1)+3\ \min(p,l^2)(\min(m,n)+1)$. CG requires one additional iteration for exact convergence.

■

In our experience, as numerical experiments show, this bound is quite loose to the extent that in practice, within at most 10 iterations or so, we have effective convergence. The two preconditioners achieve practically comparable results.

3.5 Computational Complexity Issues

Here, we make a computational cost per iteration comparison between preconditioned and unpreconditioned CG for superresolution. For unpreconditioned CG for least squares, the main computational complexity per iteration involves two matrix-vector multiplies, one with the system matrix H and one with H^T [5]. We recall from (3.4) that the matrix H is a stack of p block rows $H_k = DC_kE_k$ representing the degradation process for the LR frames. For simplicity, we assume that relative motions between frames are purely translational by integer multiples of one HR pixel or have been corrected to be so. By our spatial linear invariance assumption for the blurring kernels C_k, DE_kC_k and DC_kE_k are equivalent operators allowing us to switch the order of operations. The matrix D represents down-sampling by some factor. Thus, DE_k still represents down-sampling, but now sampling points are shifted in multiples of one HR pixel by E_k. The computational cost of convolving a kernel of size $q \times q$, with an image of $l^2N \times l^2N$ ($N = mn$, $N >> q$) is $\mathcal{O}(N\log N)$ independent of q. The cost of performing the combined operation DE_k is negligible in comparison. Therefore, the total cost per iteration of unpreconditioned CG for the least squares superresolution problem is $\mathcal{O}(pl^2N\log(l^2N))$.

In the preconditioned version, for each iteration we have an additional step of solving a linear system with the preconditioner as coefficient matrix. Because our preconditioners are derived from circulant matrices, these linear solves are performed efficiently using decomposition (3.29). This involves $2p$ FFTs of $N \times N$ blocks and a linear solve of a $p \times p$ block matrix of $N \times N$ diagonal blocks, which is $\mathcal{O}(p^2N)$. Hence, the computational complexity of the preconditioner solve is $\mathcal{O}(pN\log N)$, with $p \approx \log N$.

Computational cost arguments above show that the theoretical complexity of using our preconditioners is small compare to the total complexity for CG. In the next section, timing experiments show concrete savings achieved by using the proposed preconditioners.

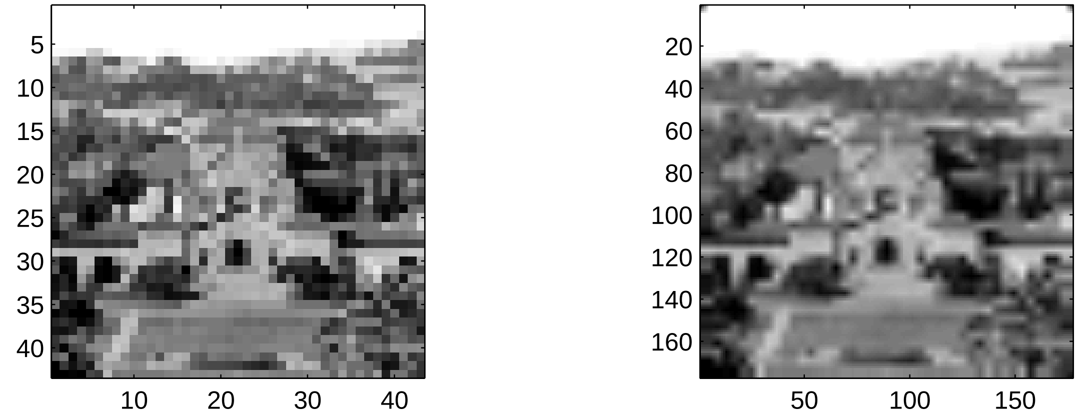
3.6 Experiments

The first test sequence consists of artificially generated LR frames. In this experiment, we compare the quality of superresolution against the original image. We shift a single 172×172 pixels image with various subpixel shifts, apply blurring with a 4×4 Gaussian PSF with standard deviation of 1 and down-sample to produce 43×43 LR frames. Using 10 of these LR frames, we produce an estimate for the original HR image. Figure 3.4 presents the results from our superresolution algorithm. The top left portion displays a sample LR image, the top right the result of bilinearly interpolating one LR frame by a factor of 4 in each dimension, the bottom left the result from superresolution after 4 PCG iterations, and the bottom right is the original image. We stop the algorithm when the relative residual[2] tolerance of 10^{-2} is reached. We use regularization parameter $\lambda = 0.001$ calculated with our approximate GCV criterion as described in Chapter 4. This experiment demonstrates the practicality of superresolution. The quality of reconstruction is comparable to that of the original image even though only a few LR frames are available. There is also a marked improvement over simple bilinear interpolation. Intuitively, this is not surprising because bilinear interpolation uses pixel values from just one frame compared to ten frames for the superresolution result.

The next two figures 3.5 and 3.6 illustrate the effectiveness of the preconditioners

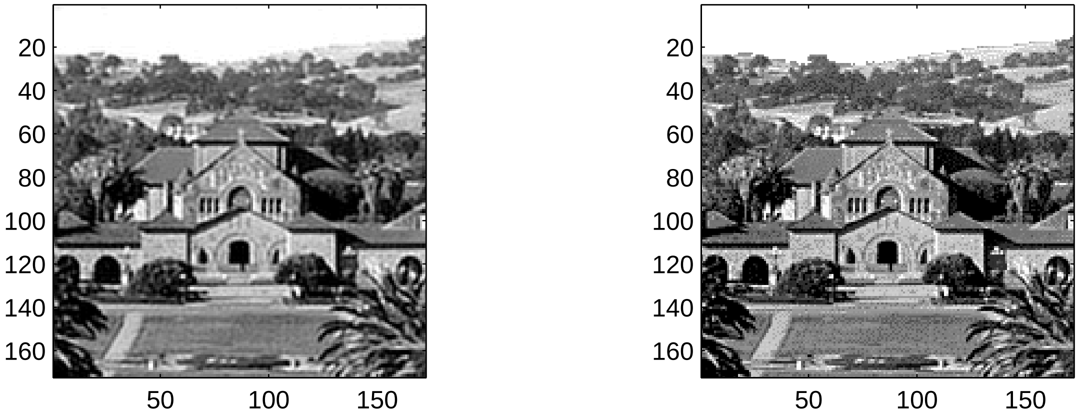
---

[2] Relative residual is defined as the ratio $\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2}$, where $\mathbf{r}_0$ is the initial residual and $\mathbf{r}_k$ is the current residual after k iterations. The residual is defined as $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$, where the vector $\mathbf{x}_k$ is the current estimate of the solution.



Sample LR frame

Bilinear interpolation



4 iterations of PCG

Original image

Figure 3.4: Superresolution results for Stanford image sequence

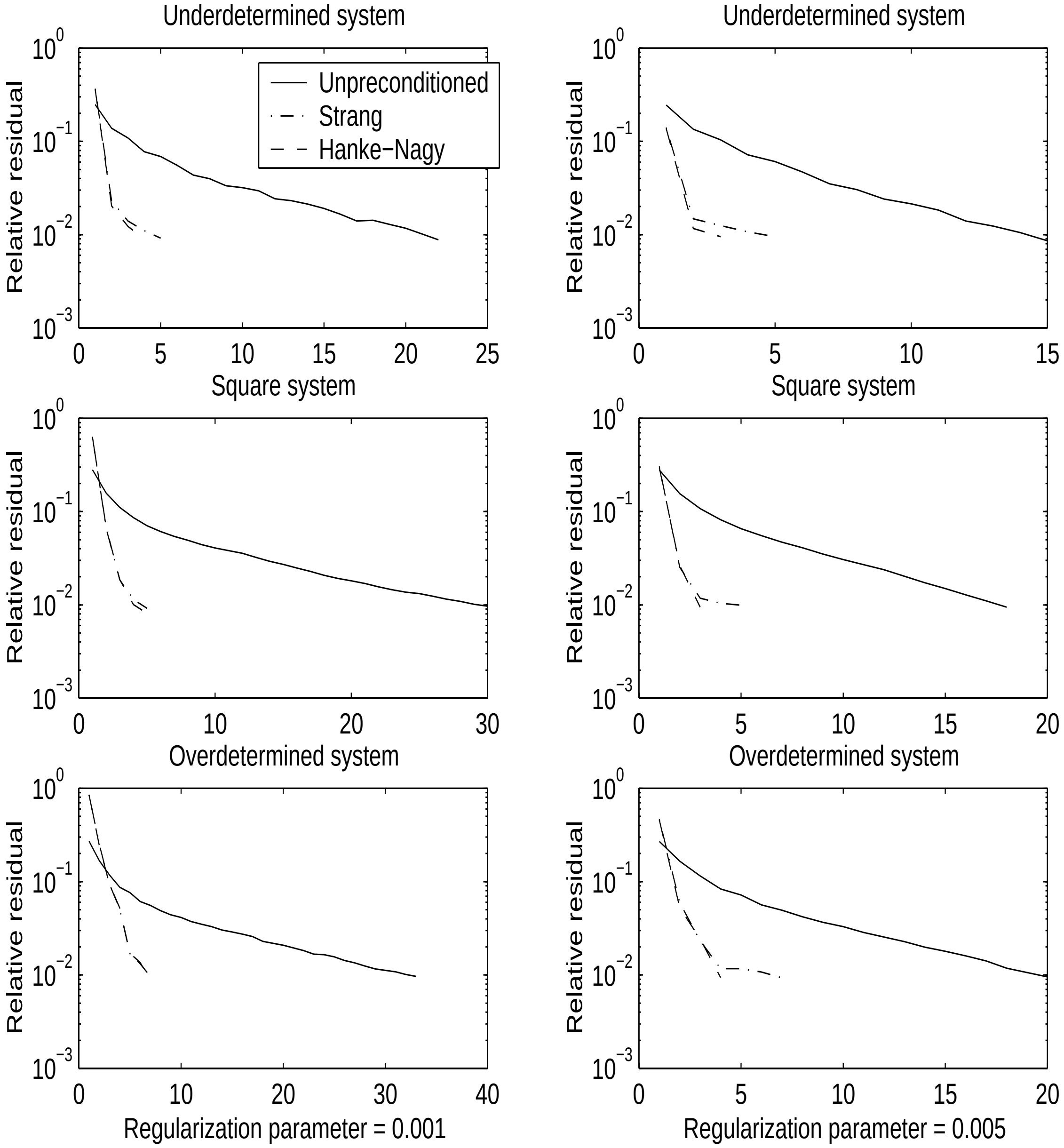


Figure 3.5: Convergence plots for Stanford sequence: $\lambda = 0.001,\ 0.005$.

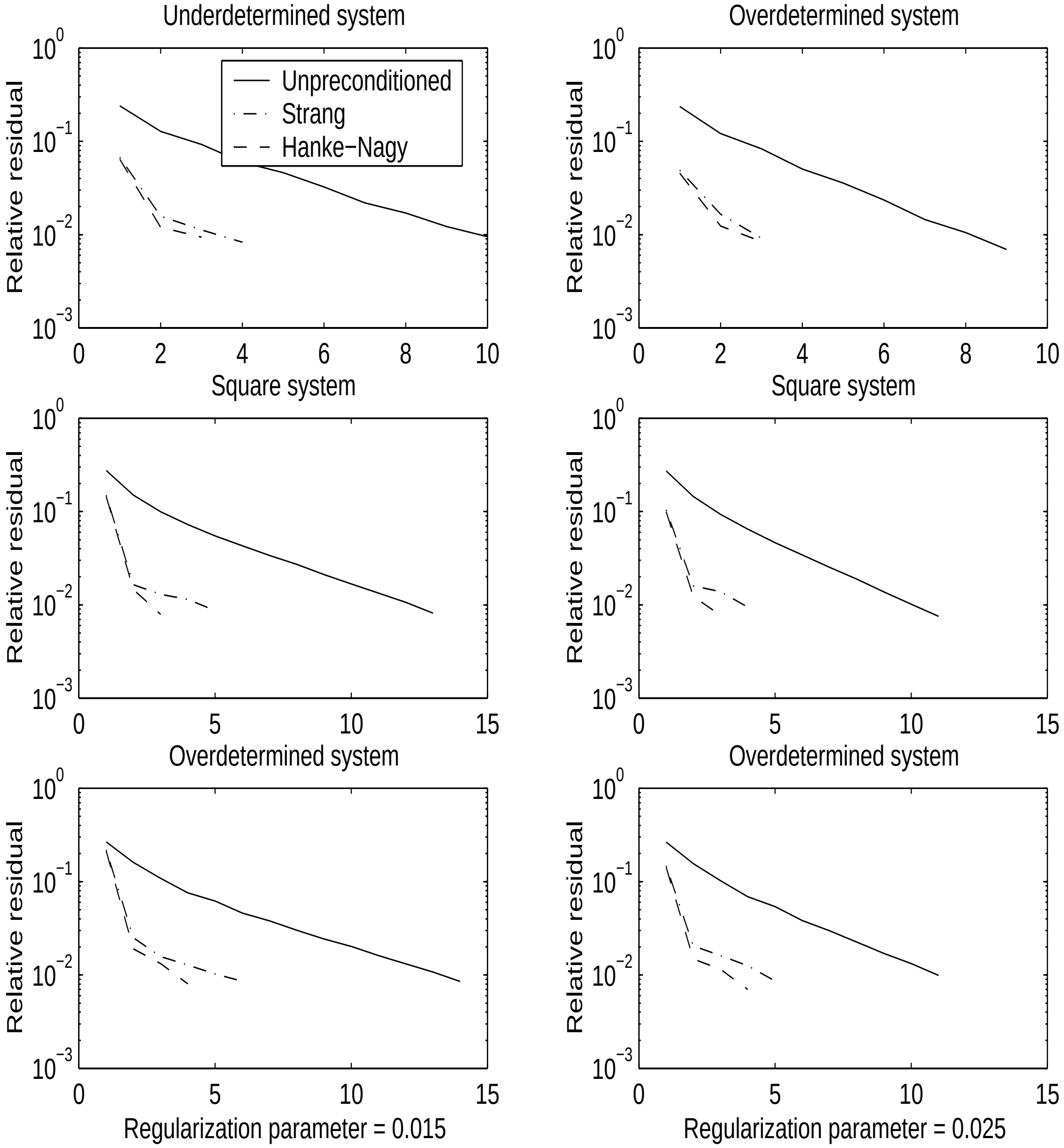


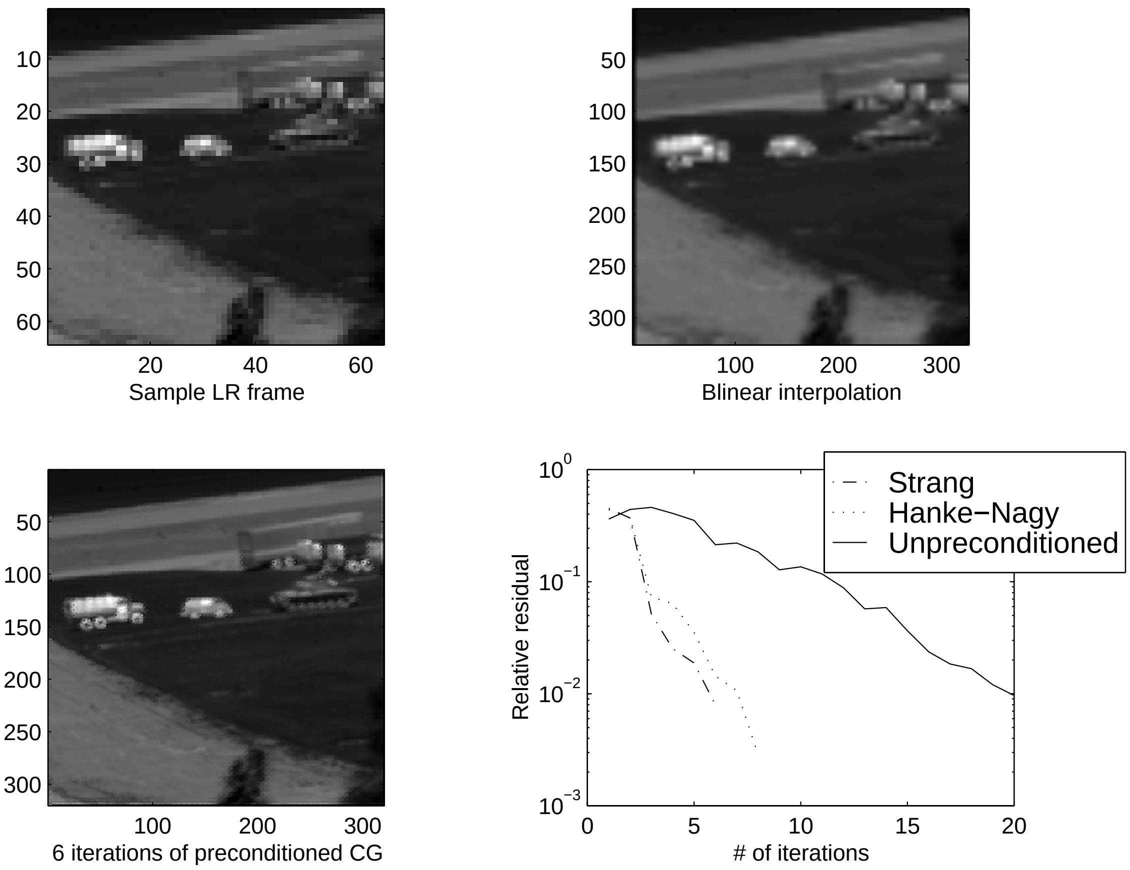
Figure 3.6: Convergence plots for Stanford sequence: $\lambda = 0.015,\ 0.025$.

	Underdetermined	Square	Overdetermined
$\lambda = 0.001$	22(99.2) 5(34.8) 5(34.3)	30(212.3) 5(55.8) 5(70.2)	33(292.1) 7(97.2) 7(122.4)
$\lambda = 0.005$	15(65.7) 5(34.2) 3(26.2)	18(132.0) 5(57.4) 3(41.8)	20(187.9) 7(103.0) 4(72.6)
$\lambda = 0.015$	10(45.8) 4(29.1) 3(27.1)	13(94.9) 5(55.4) 3(42.2)	14(128.1) 6(85.0) 4(69.4)
$\lambda = 0.025$	9(40.2) 3(20.6) 3(25.4)	11(80.9) 4(53.4) 3(44.2)	11(98.6) 5(71.2) 4(68.8)

Table 3.3: Iteration counts and timing results for preconditioned vs. unpreconditioned CG for superresolution on a Sun UltraSparc-I for the Stanford image sequence. Each box contains iteration count (timing in seconds) for unpreconditioned (top), Strang's preconditioner (middle), and Hanke-Nagy's preconditioner (bottom).

with respect to the conditioning of the system. The experiments in each column of each figure apply the same regularization parameter. We plot relative residuals as functions of the number of iterations. The top plots in each figure show the experiments for the underdetermined superresolution problem described above. In this case, we use 10 LR frames and compare the convergence rates of preconditioned and unpreconditioned CG. The middle plots in each figure contain results for the square problem, i.e. with 16 LR data frames, and the bottom plots have experiments for the overdetermined problem with 20 frames. The convergence curves for CG with the proposed preconditioners are much steeper than without. In general, the number of iterations required to reach the same tolerance level is reduced by a factor of 3. Iteration counts and timing results on a Sun UltraSparc-I are assembled in Table 3.3. The savings in iteration counts translate to roughly 50% savings in overall computation time.

The low resolution FLIR images in our second test sequence are provided courtesy of Brian Yasuda and the FLIR research group in the Sensors Technology Branch, Wright Laboratory, WPAFB, OH. Results using this data set are also shown in [39].



Sample LR frame

Blinear interpolation

6 iterations of preconditioned CG

Figure 3.7: Superresolution results for FLIR image sequence

```
$\mathbf{x}_0 = 0;$
$\beta_1 = \|\mathbf{b}\|_2; \quad \mathbf{u}_1 = \mathbf{b}/\beta_1;$
$\mathbf{v}_1 = A^T\mathbf{u}_1; \quad \alpha_1 = \|\mathbf{v}_1\|_2; \quad \mathbf{v}_1 = \mathbf{v}_1/\alpha_1;$
$\mathbf{w}_1 = \mathbf{v}_1;$
$\bar{\phi}_1 = \beta_1; \quad \bar{\rho}_1 = \alpha_1;$

for $i = 0 : max\_it$
    $\mathbf{u}_{i+1} = A\mathbf{v}_i - \alpha_i\mathbf{u}_i; \quad \beta_{i+1} = \|\mathbf{u}_{i+1}\|_2; \quad \mathbf{u}_{i+1} = \mathbf{u}_{i+1}/\beta_{i+1};$
    $\mathbf{v}_{i+1} = A^T\mathbf{u}_{i+1} - \beta_{i+1}\mathbf{v}_i; \quad \alpha_{i+1} = \|\mathbf{v}_{i+1}\|_2; \quad \mathbf{v}_{i+1} = \mathbf{v}_{i+1}/\alpha_{i+1};$
    $[c_i, s_i, \rho_i] = \text{givens\_rot}(\bar{\rho}_i, \beta_{i+1});$
    $\theta_i = s_i\alpha_{i+1}; \quad \bar{\rho}_{i+1} = c_i\alpha_{i+1};$
    $\phi_i = c_i\bar{\phi}_i; \quad \bar{\phi}_{i+1} = -s_i\bar{\phi}_i;$
    $\mathbf{x}_i = \mathbf{x}_{i-1} + (\phi_i/\rho_i)\mathbf{w}_i;$
    $\mathbf{w}_{i+1} = \mathbf{v}_{i+1} - (\theta_{i+1}/\rho_i)\mathbf{w}_i;$
end
```

Table 3.4: LSQR algorithm.

Each LR image is 64×64 pixels, and a resolution enhancement factor of 5 is sought. The objects in the scene are stationary, and 16 frames are acquired by controlled movements of a FLIR imager described in [39]. Figure 3.7 has similar arrangements as Figure 3.4 except now the bottom right shows the relative residual graphs for the unpreconditioned and preconditioned runs. For this sequence, we again set the relative residual tolerance to 10^{-2} and use regularization parameter $\lambda = 0.0076$. Six iterations are required for PCG with Strang's preconditioner and 8 iterations for Hanke and Nagy's approximate inverse preconditioner versus 22 for unpreconditioned CG, to reach the residual threshold.

3.7 LSQR Algorithm

Recall that our regularized least squares superresolution problem may be formulated as follows:

$$\min_{\mathbf{x}} \left\| \begin{pmatrix} H \\ \sqrt{\lambda} I \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix} \right\|_2^2 . \tag{3.64}$$

This suggests LSQR, developed by Paige and Saunders [67], as an alternative to CG for solving our regularized least squares superresolution. Although LSQR and CG applied to (3.64) generate mathematically the same sequence of approximations, LSQR has been shown to be numerically more stable than CG when A is ill-conditioned and many iterations are needed. For completeness, we include the pseudocode for LSQR from [12, p. 308] in Table 3.4. The procedure $[c, s, \rho] = \text{givens\_rot}(\alpha, \beta)$ is the Givens rotation that zeros out the second entry:

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \sigma \\ 0 \end{pmatrix} . \tag{3.65}$$

In [66] the algorithm is extended to implicitly handle the regularization term $\sqrt{\lambda} I$. LSQR can then be applied to (3.64) with $A = H$, $\mathbf{b} = \mathbf{f}$ and $\lambda > 0$. Similar to CG, the main computational burden for LSQR is the matrix-vector products with A and A^T. Storage requirements are also roughly the same for both. Because the numerical stability advantage of LSQR over CG is realized when many iterations are required and for our superresolution experiments only a few iterations are needed even for unpreconditioned CG, we used the slightly simpler CG algorithm.

3.8 Conclusion

In this chapter, we presented efficient preconditioners for convergence acceleration for image superresolution. Our robust approach for superresolution reconstruction employed Tikhonov regularization. To calculate the regularization parameter automatically, we adapted the generalized cross-validation criterion. We discuss regularization in more detail in the next chapter. Using a synthetic image sequence, we demonstrated that superresolution produces a high-quality estimate of the original image. To accelerate CG convergence, we proposed circulant-type preconditioners based on previous work by Strang, Hanke and Nagy. These preconditioners can be easily constructed, operations involving these preconditioners can be done efficiently by FFTs, and most importantly, the number of CG iterations is dramatically reduced as a result of preconditioning. We showed upper bounds, for both preconditioners, on the number of PCG iterations necessary for exact convergence. In practice, we observed that PCG takes at most $\frac{1}{3}$ the number of iterations of unpreconditioned CG, leading to significant improvements in runtime (roughly 50%). The savings were even better for smaller regularization parameters λ. Typically, we stopped after 5 PCG iterations because results obtained thereafter are not significantly different visually. By these experiments, we have demonstrated that with the use of appropriate preconditioners, image superresolution can be made computationally much more tractable.

Chapter 4

Regularization for Superresolution

4.1 Introduction

Much research has been done on how to solve linear ill-posed inverse problems stably. In particular, Fredholm integral equations of the first kind have been throughly studied. These equations can be expressed generally as:

$$\int_\Omega h(s,t)x(t)dt = f(s), \quad s \in \Omega \subset \mathbf{R}^q, \tag{4.1}$$

with $h(\cdot,\cdot) \in L^2(\Omega \times \Omega)$. If $h(s,t)$ is translation invariant, then $h(s,t) = h(s-t)$ and equation (4.1) becomes a convolution equation with kernel $h(s)$,

$$\begin{aligned} f(s) &= \int_\Omega h(s-t)x(t)dt & (4.2) \\ &= h(s) \ * \ x(s), & (4.3) \end{aligned}$$

where $*$ denotes the convolution operator. Let $H : L^2(\Omega) \to L^2(\Omega)$ be the linear convolution operator $Hx = h \ * \ x$. Then H is a compact operator with the following singular value expansion:

$$Hv_j = \sigma_j u_j, \quad H^* u_j = \sigma_j v_j, \quad j = 1, 2, ... \tag{4.4}$$

with H^* being the adjoint operator, σ_j a nonincreasing sequence of positive singular values and u_j, v_j the corresponding singular functions. Expanding the right-hand side f in terms of u_j we have

$$f = \sum_{j=1}^{\infty} \eta_j u_j, \ \eta_j = (u_j, f), \tag{4.5}$$

where $(\cdot,\cdot)$ denotes the inner product. The solution $x = H^{-1}f$ converges only if f satisfies the Picard condition [37]:

$$\sum_{j=1}^{\infty} \left(\frac{\eta_j}{\sigma_j}\right)^2 < \infty. \tag{4.6}$$

In practice, the right hand side f contains some amount of noise, e.g. from measurement errors, modeling errors, etc. Hence, solving for x directly is fruitless, as even small changes in f can result in wild oscillations in approximations to x. Recall from Chapter 3 the system of equations for superresolution:

$$\begin{aligned} \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_p \end{bmatrix} &= \begin{bmatrix} DC_1E_1 \\ \vdots \\ DC_pE_p \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{n}_1 \\ \vdots \\ \mathbf{n}_p \end{bmatrix} \\ \mathbf{f} &= H\mathbf{x} + \mathbf{n}. \end{aligned} \tag{4.7}$$

The operators C_i's are discretized blurring operators, (i.e. 2-D convolutions). Therefore, H is an ill-conditioned matrix [4]. To obtain a reasonable estimate for $\mathbf{x}$, we need to “regularize”, i.e. enforce additional constraints that favor well-behaved solutions. Tikhonov first introduced the regularization concept for square/overdetermined least squares problems in his seminal paper [84]. For noisy overdetermined systems, we search for solutions to fit to the noisy data

$$\min_{\mathbf{x}} \ \|H\mathbf{x} - \mathbf{f}\|_2^2 + \lambda \|L\mathbf{x}\|_2^2, \tag{4.8}$$

where L is a regularization operator, and λ is related to the Lagrange multiplier. The first term in (4.8) ensures that the estimated solution has small residuals, while the second favors “well-behaved” solutions. With L being the identity, (4.8) encourages solutions with small 2-norm. The matrix L can also incorporate prior knowledge of the problem, e.g. degree of smoothness [37]. The scalar λ serves as a balance between two requirements. On one hand, we would like the regularized solution to have small residual norm $\|H\mathbf{x} - \mathbf{f}\|_2$. On the other hand, it should be a well-behaved solution. The scalar λ plays a delicate role. If λ is chosen too large, the regularized system is too far away from the original equation. If λ is too small, then the regularized system would still be ill-conditioned. Figure 4.1 illustrates the importance of choosing an appropriate regularization parameter. We show superresolution results using various levels of regularization. The under-regularized solution is overwhelmed with noise

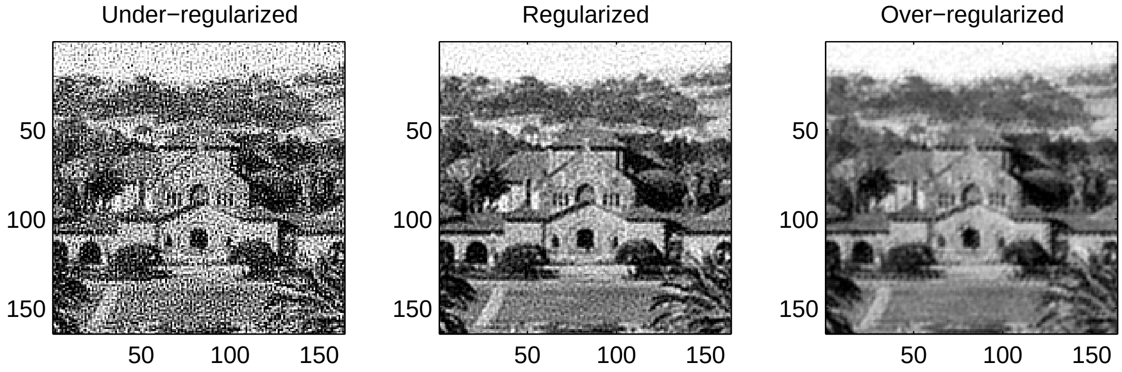


Figure 4.1: Superresolution reconstructions with various levels of regularization.

and registration artifacts. On the other hand, the over-regularized solution smoothes out some scene details.

Differentiating the objective function in (4.8) with respect to $\mathbf{x}$ and setting to 0, we get

$$\mathbf{x}(\lambda) = (H^T H + \lambda L^T L)^{-1} H^T \mathbf{f}. \tag{4.9}$$

The regularized solution $\mathbf{x}(\lambda)$ depends on the regularization parameter λ. For square or overdetermined least squares systems, regularization parameter estimation is a well-studied area of research. In this chapter, we examine regularization parameter estimation for *underdetermined* least squares problems, which has received very little attention in the literature.

Our approach to regularizing the underdetermined superresolution follows the commonly used least squares formulation for underdetermined systems:

$$\min_{\mathbf{x}} \ \alpha \|H\mathbf{x} - \mathbf{f}\|_2^2 + \|L\mathbf{x}\|_2^2. \tag{4.10}$$

In this instance, the regularization parameter is in front of the residual term instead of the penalty term as in (4.8). However, the two formulae, (4.8) and (4.10), are equivalent with $\alpha = 1/\lambda$. As Saunders mentioned in [74], the least squares solutions of (4.8) and (4.10) for $L = I$ are both solved by the augmented system

$$\begin{pmatrix} \sqrt{\lambda} I & H \\ H^T & -\sqrt{\lambda} I \end{pmatrix} \begin{pmatrix} \mathbf{s} \\ \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{0} \end{pmatrix}. \tag{4.11}$$

In (4.10), L represents constraints on the solution $\mathbf{x}$, so we assume it to be of full row rank and either square or underdetermined. We first transform the general regularization formula (4.10) into standard form, (i.e. with $L = I$), in the next subsection. All regularization techniques described subsequently assume L to be the identity for convenience. Closed-form solutions for regularized least squares for underdetermined systems and non-Tikhonov regularization techniques are briefly discussed in Subsections 4.1.2 and 4.1.3, respectively.

4.1.1 Transforming to Standard Form

If L is square, let $\tilde{\mathbf{x}} = L\mathbf{x}, A = HL^{-1}$. Then (4.10) is equivalent to:

$$\min_{\tilde{\mathbf{x}}} \alpha \|A\tilde{\mathbf{x}} - \mathbf{f}\|_2^2 + \|\tilde{\mathbf{x}}\|_2^2. \tag{4.12}$$

The more complicated situation is when L is underdetermined. The following derivation closely resembles work in Haber's thesis [36]. Suppose that the dimensions of H are $M \times N$, and the dimensions of L are $K \times N$, with both $M, K < N$. Since both H and L are of full rank, equation (4.10) has unique solution only if $\mathcal{N}(H) \cap \mathcal{N}(L) = \mathbf{0}$. This occurs only if $M + K \geq N$. Let L_0 be some $N \times (N - K)$ matrix representing the null space of L. Any vector $\mathbf{x}$ can be partitioned into two parts, one belonging to null space of L and one in the space orthogonal to the null space, the so-called active space of L:

$$\begin{aligned} \mathbf{x} &= \mathbf{x}_a + \mathbf{x}_0 \\ &= \mathbf{x}_a + L_0 \tilde{\mathbf{x}}_0. \end{aligned} \tag{4.13}$$

Elden [26] has noted that vectors in the active space can be written as

$$\mathbf{x}_a = L_A^+\tilde{\mathbf{x}}, \tag{4.14}$$

where $L_A^+ = L^+ - L_0(AL_0)^+AL^+$ is the A-weighted generalized inverse of L. Equation (4.10) can be reformulated in matrix form as the least squares solution to

$$\begin{bmatrix} \sqrt{\alpha}A \\ L \end{bmatrix} \left[L_A^+\tilde{\mathbf{x}} + L_0\tilde{\mathbf{x}}_0\right] = \begin{bmatrix} \sqrt{\alpha}\mathbf{f} \\ \mathbf{0} \end{bmatrix}. \tag{4.15}$$

Since $LL_0 = 0$,

$$\begin{aligned} LL_A^+ &= L(L^+ - L_0(AL_0)^+AL^+) \\ &= LL^+ \\ &= I. \end{aligned} \tag{4.16}$$

We can rewrite the equation for $\tilde{\mathbf{x}}_0$ in (4.15) as

$$AL_0\tilde{\mathbf{x}}_0 = \mathbf{f} - AL_A^+\tilde{\mathbf{x}}. \tag{4.17}$$

Note that AL_0 has dimensions $M \times (N-K)$, so it must be square or overdetermined. Hence, we can solve for $\tilde{\mathbf{x}}_0$ in the equation above:

$$\tilde{\mathbf{x}}_0 = (AL_0)^+\mathbf{f} - (AL_0)^+AL_A^+\tilde{\mathbf{x}}. \tag{4.18}$$

Using the expanded form for L_A^+ we get

$$\begin{aligned} (AL_0)^+AL_A^+ &= (AL_0)^+A(L^+ - L_0(AL_0)^+AL^+) \\ &= (AL_0)^+AL^+ - (AL_0)^+AL_0(AL_0)^+AL^+ \\ &= (AL_0)^+AL^+ - (AL_0)^+AL^+ \\ &= 0. \end{aligned} \tag{4.19}$$

Equation (4.18) gives the solution for $\tilde{\mathbf{x}}_0$ independent of $\tilde{\mathbf{x}}$:

$$\tilde{\mathbf{x}}_0 = (AL_0)^+\mathbf{f}. \tag{4.20}$$

Substituting this into (4.15), we obtain an equation in standard form:

$$\begin{bmatrix} \sqrt{\alpha} A L_A^+ \\ I \end{bmatrix} \tilde{\mathbf{x}} = \begin{bmatrix} \sqrt{\alpha}(I - AL_0(AL_0)^+)\mathbf{f} \\ \mathbf{0} \end{bmatrix}. \tag{4.21}$$

4.1.2 Closed-form Solution for Underdetermined Systems

In the previous subsection, we have transformed our problem into standard form:

$$\min_{\mathbf{x}} \alpha \|A\mathbf{x} - \mathbf{f}\|_2^2 + \|\mathbf{x}\|_2^2. \tag{4.22}$$

Differentiating with respect to $\mathbf{x}$ and setting to 0, we get:

$$(I + \alpha A^T A)\mathbf{x} = \alpha A^T \mathbf{f}, \tag{4.23}$$

$$\mathbf{x} = \alpha (I + \alpha A^T A)^{-1} A^T \mathbf{f} \tag{4.24}$$

We make use of the following relation:

$$I - M^T(MM^T + cI)^{-1}M = c(M^T M + cI)^{-1} \tag{4.25}$$

for any matrix M and scalar $c > 0$. For $M = A, c = \frac{1}{\alpha}$,

$$\begin{aligned} (A^T A + \tfrac{1}{\alpha} I)^{-1} A^T &= \alpha (I - A^T (AA^T + \tfrac{1}{\alpha} I)^{-1} A) A^T \\ &= \alpha A^T (I - (AA^T + \tfrac{1}{\alpha} I)^{-1} AA^T) \\ &= A^T (AA^T + \tfrac{1}{\alpha} I)^{-1}. \end{aligned} \tag{4.26}$$

Hence,

$$\mathbf{x} = A^T(AA^T + \lambda I)^{-1}\mathbf{f}, \quad \lambda = \frac{1}{\alpha}. \tag{4.27}$$

This is the least squares solution for the Tikhonov regularized underdetermined problem (4.22).

4.1.3 Other Regularization Techniques

In this subsection, we briefly review alternatives to Tikhonov regularization for solving noisy linear inverse problem. These techniques fall mainly into three distinct categories: low-pass filtering techniques, iterative regularization methods, and statistically-based regularization methods.

Low-pass filtering is based on the assumption that most of the energy of the true solution of the system is contained in a few low frequencies, while the noise energy is evenly distributed throughout. Low-pass filtering methods attenuate the high frequency components, eliminating most of the noise effects and fine scale details of the solution at the same time. Truncated SVD is a low-pass filtering example where the components in the solution corresponding to the smallest singular values are eliminated [12]. Iterative methods are also known to have regularization properties. Some iterative methods, such as Kaczmarz's method [49] and CG, exhibit semiconvergence behavior when applied to a noisy linear system, where the residual decreases for a few iterations then diverges. By choosing appropriately where to stop the iterations, we obtain solutions minimally contaminated by noise (cf. [37]). Statistical methods based on the maximum entropy criterion and Bayes' rule, e.g. distribution-entropy, maximum likelihood (ML), maximum a posteriori (MAP) [15, 86], are often used for regularized restoration in the engineering community. They rely on statistical models for observed images to predict estimates for the unperturbed image.

4.2 Regularization Criteria

In this section, we describe two methods for automatically calculating the regularization parameter in (4.10). The regularization parameter plays an important role in the formulation of a well-conditioned problem. If α is chosen too large, the regularization penalty term will be insignificant, and the system remains ill-conditioned. On the other hand, if α is too small, the regularized solution will have large residuals. All methods described below are data-driven and do not need noise level estimates. Data-driven, unsupervised algorithms are necessary in many practical applications, where sometimes even the noise model itself is not known.

4.2.1 Quasi-optimality Criterion

The quasi-optimality criterion was first introduced by Tikhonov and Glasko [85] in the continuous setting. The intuition behind the method is that near the optimal regularization parameter α_q, the norm of the difference between the regularized solution $\mathbf{x}_\alpha$ and the exact solution $\mathbf{x}_{\text{exact}}$ should form a plateau. We use the same intuition to derive a new and slightly different form of the quasi-optimality criterion for underdetermined systems. The regularization parameter is chosen to minimize (cf. [37])

$$\phi_Q(\alpha) = \left\| \alpha \frac{d\mathbf{x}_\alpha}{d\alpha} \right\|_2^2, \quad \alpha > 0, \tag{4.28}$$

where for least squares for underdetermined systems, the regularized solution $\mathbf{x}_\alpha$ is given by (4.27)

$$\mathbf{x}_\alpha = A^T(AA^T + \frac{1}{\alpha} I)^{-1}\mathbf{f}. \tag{4.29}$$

We use the following property of matrix differentiation [33]:

$$\frac{d}{d\alpha}\left[M(\alpha)^{-1}\right] = -M(\alpha)^{-1}\left[\frac{d}{d\alpha} M(\alpha)\right] M(\alpha)^{-1}. \tag{4.30}$$

Thus,

$$\begin{aligned} \frac{d\mathbf{x}_\alpha}{d\alpha} &= A^T(AA^T + \frac{1}{\alpha}I)^{-1}(-\frac{1}{\alpha^2}I)(AA^T + \frac{1}{\alpha}I)^{-1}\mathbf{f} && (4.31) \\ &= -\frac{1}{\alpha^2}A^T(AA^T + \frac{1}{\alpha}I)^{-2}\mathbf{f}. && (4.32) \end{aligned}$$

The regularization parameter minimizes the quasi-optimality criterion

$$\phi_Q(\alpha) = \frac{1}{\alpha^2}\mathbf{f}^T(AA^T + \frac{1}{\alpha}I)^{-2}AA^T(AA^T + \frac{1}{\alpha}I)^{-2}\mathbf{f}. \qquad (4.33)$$

In terms of $\lambda = \frac{1}{\alpha}$,

$$\begin{aligned} \phi_Q(\lambda) &= \lambda^2\mathbf{f}^T(AA^T + \lambda I)^{-2}AA^T(AA^T + \lambda I)^{-2}\mathbf{f} && (4.34) \\ &= \lambda^2\mathbf{f}^T(AA^T + \lambda I)^{-3}\mathbf{f} - \lambda^3\mathbf{f}^T(AA^T + \lambda I)^{-4}\mathbf{f}. && (4.35) \end{aligned}$$

Clearly, $\lambda = 0$ is a global minimizer of (4.34). The desired local minimizer of $\phi_Q(\lambda)$ is to the right of $\lambda = 0$.

4.2.2 Generalized Cross-validation

Regularization parameter estimation using cross-validation was first introduced by Wahba [90, 30]. The idea of cross-validation is simple. For some regularization parameter estimate λ, we divide the data into two sets; one set is used to construct an approximate solution based on λ, and the other is used to measure the error of that approximation. For example, the validation error by using the jth pixel value as the validation set is:

$$CV_{\cancel{j}}(\lambda) = \|\mathbf{a}_j\mathbf{x}_{\cancel{j}}(\lambda) - f_j\|_2^2, \qquad (4.36)$$

where the terms with crossed-out j are associated with the system without the jth pixel, i.e.,

$$\mathbf{x}_{\cancel{j}}(\lambda) = A_{\cancel{j}}^T(A_{\cancel{j}}A_{\cancel{j}}^T + \lambda I)^{-1}\mathbf{f}_{\cancel{j}} \qquad (4.37)$$

is the regularized least squares solution of the original underdetermined system without the jth pixel, $A_{\cancel{j}}\mathbf{x}_{\cancel{j}} = \mathbf{f}_{\cancel{j}}$, with

$$A_{\cancel{j}} = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_{j-1} \\ \mathbf{a}_{j+1} \\ \vdots \\ \mathbf{a}_K \end{bmatrix}, \quad \mathbf{f}_{\cancel{j}} = \begin{bmatrix} f_1 \\ \vdots \\ f_{j-1} \\ f_{j+1} \\ \vdots \\ f_K \end{bmatrix}. \tag{4.38}$$

We rotate through all pixels, using each in turn as the validation set, and sum the validation errors. The optimal regularization parameter λ_{CV} minimizes the total validation error:

$$\lambda_{CV} = \text{argmin}_{\lambda} \sum_{j=1}^{K} CV_{\cancel{j}}(\lambda). \tag{4.39}$$

Generalized cross-validation (GCV) is simply cross-validation applied to the original system after it has undergone a unitary transformation. GCV is also known to be less sensitive to large individual equation errors (outliers) than cross-validation [61]. For overdetermined systems, it has been shown that the asymptotically optimum regularization parameter according to GCV is given by [30]:

$$\lambda_{GCV} = \text{argmin}_{\lambda} \frac{\|(AA^T + \lambda I)^{-1}\mathbf{f}\|_2}{tr((AA^T + \lambda I)^{-1})}. \tag{4.40}$$

GCV is well-known for calculating regularization parameters for Tikhonov-regularized *overdetermined* least squares problems without accurate knowledge of the variance of noise. Golub, Heath, and Wahba [30] suggested that GCV can be used for underdetermined problems, and McIntosh and Veronis [61] successfully implemented GCV for their underdetermined tracer inverse problems. However, since the derivation for the GCV formulation by Golub, Heath, and Wahba [30] applies only for overdetermined least squares problems, we extend their derivation to the underdetermined case in the next section.

4.3 GCV for Underdetermined Systems

Here we derive the closed-form for regularization parameter by GCV for *underdetermined* systems $A\mathbf{x} = \mathbf{f}$. We start by examining standard cross-validation with the following expression for the regularized least squares solution to the underdetermined equation above:

$$\begin{aligned}
\mathbf{x} &= A^T(AA^T + \lambda I)^{-1}\mathbf{f} && (4.41)\\
&= \begin{bmatrix} A_{\not{j}}^T & \mathbf{a}_j^T \end{bmatrix} \left(\begin{bmatrix} A_{\not{j}} \\ \mathbf{a}_j \end{bmatrix} \begin{bmatrix} A_{\not{j}}^T & \mathbf{a}_j^T \end{bmatrix} + \lambda I \right)^{-1} \begin{bmatrix} \mathbf{f}_{\not{j}} \\ f_j \end{bmatrix} && (4.42)\\
&= \begin{bmatrix} A_{\not{j}}^T & \mathbf{a}_j^T \end{bmatrix} \begin{bmatrix} A_{\not{j}} A_{\not{j}}^T + \lambda I & A_{\not{j}} \mathbf{a}_j^T \\ \mathbf{a}_j A_{\not{j}}^T & \mathbf{a}_j \mathbf{a}_j^T + \lambda I \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_{\not{j}} \\ f_j \end{bmatrix}. && (4.43)
\end{aligned}$$

Let

$$\begin{bmatrix} \mathbf{z}_{\not{j}} \\ z_j \end{bmatrix} = \begin{bmatrix} A_{\not{j}} A_{\not{j}}^T + \lambda I & A_{\not{j}} \mathbf{a}_j^T \\ \mathbf{a}_j A_{\not{j}}^T & \mathbf{a}_j \mathbf{a}_j^T + \lambda I \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{f}_{\not{j}} \\ f_j \end{bmatrix}. \tag{4.44}$$

Then

$$\mathbf{x} = A_{\not{j}}^T \mathbf{z}_{\not{j}} + \mathbf{a}_j^T z_j, \tag{4.45}$$

and

$$\begin{aligned}
(A_{\not{j}} A_{\not{j}}^T + \lambda I)\mathbf{z}_{\not{j}} + A_{\not{j}} \mathbf{a}_j^T z_j &= \mathbf{f}_{\not{j}}, && (4.46)\\
\mathbf{a}_j A_{\not{j}}^T \mathbf{z}_{\not{j}} + (\mathbf{a}_j \mathbf{a}_j^T + \lambda I) z_j &= f_j. && (4.47)
\end{aligned}$$

From (4.47) and (4.45) we get

$$\mathbf{a}_j \mathbf{x} + \lambda z_j = f_j. \tag{4.48}$$

From (4.46) we get

$$(A_{\not{j}} A_{\not{j}}^T + \lambda I)^{-1} \mathbf{f}_{\not{j}} = \mathbf{z}_{\not{j}} + (A_{\not{j}} A_{\not{j}}^T + \lambda I)^{-1} A_{\not{j}} \mathbf{a}_j^T z_j. \tag{4.49}$$

Recall the equation for the regularized least squares for underdetermined systems:

$$\mathbf{x}_{\not{j}} = A_{\not{j}}^T (A_{\not{j}} A_{\not{j}}^T + \lambda I)^{-1} \mathbf{f}_{\not{j}}. \tag{4.50}$$

From (4.45), (4.50) and (4.49),

$$\begin{aligned} \mathbf{x}_{\not{j}} &= A_{\not{j}}^T \mathbf{z}_{\not{j}} + A_{\not{j}}^T (A_{\not{j}} A_{\not{j}}^T + \lambda I)^{-1} A_{\not{j}} \mathbf{a}_j^T z_j \\ &= \mathbf{x} - (I - A_{\not{j}}^T (A_{\not{j}} A_{\not{j}}^T + \lambda I)^{-1} A_{\not{j}}) \mathbf{a}_j^T z_j \\ &= \mathbf{x} - \lambda (A_{\not{j}}^T A_{\not{j}} + \lambda I)^{-1} \mathbf{a}_j^T z_j. \end{aligned} \tag{4.51}$$

Next, note that

$$A_{\not{j}}^T A_{\not{j}} = A^T A - \mathbf{a}_j^T \mathbf{a}_j. \tag{4.52}$$

So by the Sherman-Morrison-Woodbury formula [33],

$$\begin{aligned} (A_{\not{j}}^T A_{\not{j}} + \lambda I)^{-1} &= (A^T A + \lambda I)^{-1} + \\ &\quad (A^T A + \lambda I)^{-1} \mathbf{a}_j^T (1 - d_j)^{-1} \mathbf{a}_j (A^T A + \lambda I)^{-1}, \end{aligned} \tag{4.53}$$

where $d_j = \mathbf{a}_j (A^T A + \lambda I)^{-1} \mathbf{a}_j^T$. Therefore,

$$\begin{aligned} \mathbf{a}_j (A_{\not{j}}^T A_{\not{j}} + \lambda I)^{-1} \mathbf{a}_j^T &= d_j + d_j (1 - d_j)^{-1} d_j \\ &= d_j (1 - d_j)^{-1}. \end{aligned} \tag{4.54}$$

Hence from (4.51) and (4.54),

$$
\begin{aligned}
\mathbf{a}_j\mathbf{x}_{\not{j}} - f_j &= \mathbf{a}_j\mathbf{x} - f_j - \lambda\mathbf{a}_j(A_{\not{j}}^T A_{\not{j}} + \lambda I)^{-1}\mathbf{a}_j^T z_j \\
&= -\lambda z_j - \lambda\mathbf{a}_j(A_{\not{j}}^T A_{\not{j}} + \lambda I)^{-1}\mathbf{a}_j^T z_j \\
&= -\lambda\left(d_j(1-d_j)^{-1} + 1\right) z_j \qquad (4.55) \\
&= -\lambda(1-d_j)^{-1} z_j \\
&= (1-d_j)^{-1}(\mathbf{a}_j\mathbf{x} - f_j).
\end{aligned}
$$

Define $D = \operatorname{diag}(I - A(A^TA+\lambda I)^{-1}A^T)$ to be the diagonal matrix with the same diagonal entries as $I - A(A^TA+\lambda I)^{-1}A^T$. Note that for $\lambda > 0$, $I - A(A^TA+\lambda I)^{-1}A^T = \frac{1}{\lambda}(AA^T + \lambda I)^{-1}$. Then from (4.39), the optimal cross-validation regularization parameter is

$$
\begin{aligned}
\operatorname{argmin}_\lambda \sum_{j=1}^{K} CV_j(\lambda) &= \operatorname{argmin}_\lambda \sum_{j=1}^{K} \|\mathbf{a}_j\mathbf{x}_{\not{j}} - f_j\|_2^2 \\
&= \operatorname{argmin}_\lambda \sum_{j=1}^{K} \|(1-d_j)^{-1}(\mathbf{a}_j\mathbf{x} - f_j)\|_2^2 \\
&= \operatorname{argmin}_\lambda \|D^{-1}(AA^T(AA^T+\lambda I)^{-1} - I)\mathbf{f}\|_2^2 \\
&= \operatorname{argmin}_\lambda \lambda^2 \|D^{-1}(AA^T+\lambda I)^{-1}\mathbf{f}\|_2^2. \qquad (4.56)
\end{aligned}
$$

We have just derived the matrix formulation of cross-validation for underdetermined systems. Generalized cross-validation is simply a rotation-invariant form of cross-validation. Following [30], consider the singular value decomposition of A:

$$A = U\Sigma V^T. \qquad (4.57)$$

Now let W be the matrix representing the Fourier transform, that is,

$$(W)_{jk} = \frac{1}{\sqrt{n}} e^{\frac{2\pi ijk}{n}}, \quad j,k = 1,2,\ldots,n. \qquad (4.58)$$

The GCV estimation for regularization parameter λ can be thought of as cross-validation on the transformed system:

$$\begin{aligned} \tilde{\mathbf{f}} &= WU^T\mathbf{f} \\ &= W\Sigma V^T\mathbf{x} \\ &= \tilde{A}\mathbf{x}. \end{aligned} \tag{4.59}$$

Note that $\tilde{A}\tilde{A}^T + \lambda I = W(\Sigma\Sigma^T + \lambda I)W^T$ is a circulant matrix. Hence, $\tilde{D} = \lambda \mathrm{diag}((\tilde{A}\tilde{A}^T + \lambda I)^{-1})$ is a multiple of the identity. Thus,

$$\begin{aligned} \tilde{D} &= \tfrac{\lambda}{m}\mathrm{tr}(\tilde{A}\tilde{A}^T + \lambda I)^{-1})I \\ &= \tfrac{\lambda}{m}\mathrm{tr}((AA^T + \lambda I)^{-1})I. \end{aligned} \tag{4.60}$$

From (4.59),

$$(\tilde{A}\tilde{A}^T + \lambda I)^{-1}\tilde{\mathbf{f}} = WU^T(AA^T + \lambda I)^{-1}\mathbf{f}. \tag{4.61}$$

Hence, from (4.56) we can formulate GCV as follows:

$$\begin{aligned} \mathrm{argmin}_\lambda \lambda^2 \|\tilde{D}^{-1}(\tilde{A}\tilde{A}^T + \lambda I)^{-1}\tilde{\mathbf{f}}\|_2 &= \mathrm{argmin}_\lambda \frac{\|(\tilde{A}\tilde{A}^T + \lambda I)^{-1}\tilde{\mathbf{f}}\|_2}{\mathrm{tr}((AA^T + \lambda I)^{-1})} \\ &= \mathrm{argmin}_\lambda \frac{\|WU^T(AA^T + \lambda I)^{-1}\mathbf{f}\|_2}{\mathrm{tr}((AA^T + \lambda I)^{-1})} \\ &= \mathrm{argmin}_\lambda \frac{\|(AA^T + \lambda I)^{-1}\mathbf{f}\|_2}{\mathrm{tr}((AA^T + \lambda I)^{-1})}. \end{aligned} \tag{4.62}$$

Not surprisingly, this formulation has the same form as that of the overdetermined case.

4.4 Experiments

We compile here experiments for regularization parameter estimation using the quasi-optimality criterion and GCV. We generate 10 synthetic LR frames from the

Stanford image as before and superresolve an estimate of the original image. We simulate typical noise conditions by imposing image registration errors and additive white noise. Let $\mathbf{x}$ be the vector of pixel intensities of the original Stanford image ordered rowwise, $\mathbf{f}$ be vector of pixel values of the LR frames, and the matrix H be the system matrix representing the degradation process from HR image to LR frames. As discussed in the beginning of Chapter 3, H is dependent on the camera PSF and registration parameters. The experiments below regularize noisy systems of the form

$$\mathbf{f} = (H + E)\mathbf{x} + \mathbf{n}, \tag{4.63}$$

where the error matrix E represents misregistration effects, and $\mathbf{n}$ is a vector of cumulative system noise. We impose registration errors on all LR frames in fractions of a pixel, and model the vector $\mathbf{n}$ as Gaussian white noise. Tables 4.1 and 4.2 below tabulate the calculated regularization parameters using the quasi-optimality criterion and GCV, respectively, under various noise and misregistration conditions. As expected, with higher noise levels, both GCV and quasi-optimality criterion return larger regularization parameters. GCV tends to give values roughly in the same range for each fixed noise level, while quasi-optimality produces increasing values as motion error increases. Overall, the quasi-optimality criterion results are about 3.5–4.5 times the values calculated by GCV. Next, we compare Tikhonov regularization with GCV and quasi-optimality criterion with a data-driven, off-the-shelf wavelet denoising approach. Table 4.3 gives a quantitative comparison of quasi-optimality criterion, GCV regularization, and MATLAB's WDENCMP wavelet denoising subroutine. We use Daubechies' 4-tap wavelets [22] to decompose the unregularized approximate solution. We employ soft threshold with an adaptive thresholding scheme using Stein's unbiased risk estimate (cf. [45]). The table gives reconstruction MSE values for the three methods, with wavelet denoising at the top, quasi-optimality criterion in the middle, and GCV on the bottom. In terms of MSE, GCV is the clear winner of the three. Qualitatively, both quasi-optimality criterion and GCV produce excellent reconstruction results though quasi-optimality criterion

	Noise standard deviations					
Motion error	0	2	4	6	8	10
0	0.0010	0.0074	0.0206	0.0332	0.0453	0.0574
0.05	0.0035	0.0092	0.0212	0.0333	0.0453	0.0575
0.10	0.0175	0.0139	0.0299	0.0394	0.0494	0.0599
0.15	0.0170	0.0192	0.0292	0.0394	0.0503	0.0617

Table 4.1: Regularization for Stanford image sequence calculated from quasi-optimality criterion

	Noise standard deviations					
Motion error	0	2	4	6	8	10
0	0.0010	0.0019	0.0056	0.0098	0.0145	0.0196
0.05	0.0013	0.0020	0.0052	0.0090	0.0134	0.0183
0.10	0.0025	0.0036	0.0063	0.0100	0.0144	0.0193
0.15	0.0033	0.0049	0.0060	0.0092	0.0131	0.0174

Table 4.2: Regularization for Stanford image sequence calculated from generalized cross-validation

tends to slightly over-regularize, i.e. the HR reconstruction oversmooths some details. Wavelet denoising with adaptive thresholding generally performs poorly with much noise remains unfiltered. Figure 4.2 compares typical superresolution reconstructions using GCV and quasi-optimality regularization techniques. The desired enhancement factor is 4. Ten sample LR frames in this experiment are degraded by 0.1 pixels motion error and noise realizations of 21 dB. The regularization parameters used were 0.0394 as calculated by the quasi-optimality criterion and 0.01 by GCV. We ran a similar experiment with our FLIR data sequence. Figure 4.3 compares regularized superresolution reconstructions with wavelet denoising at the top right corner, GCV regularization at the bottom left, and quasi-optimal regularization at the bottom right. In this case, sixteen LR frames are used to superresolve a factor of 5 enhancement. We draw similar conclusions from this experiment as with the previous experiment. While both GCV and quasi-optimality criterion produce excellent quality reconstruction, quasi-optimality slightly over-regularize.

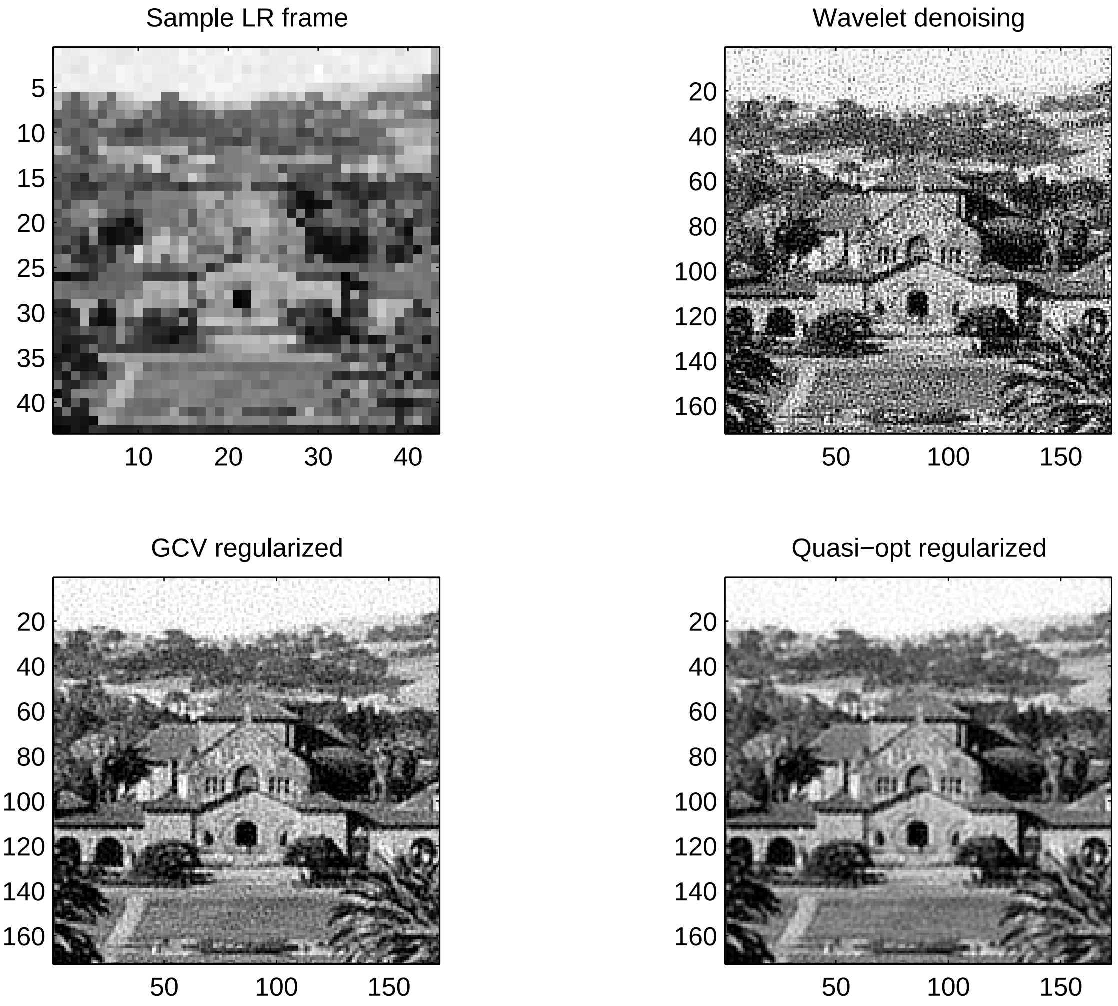
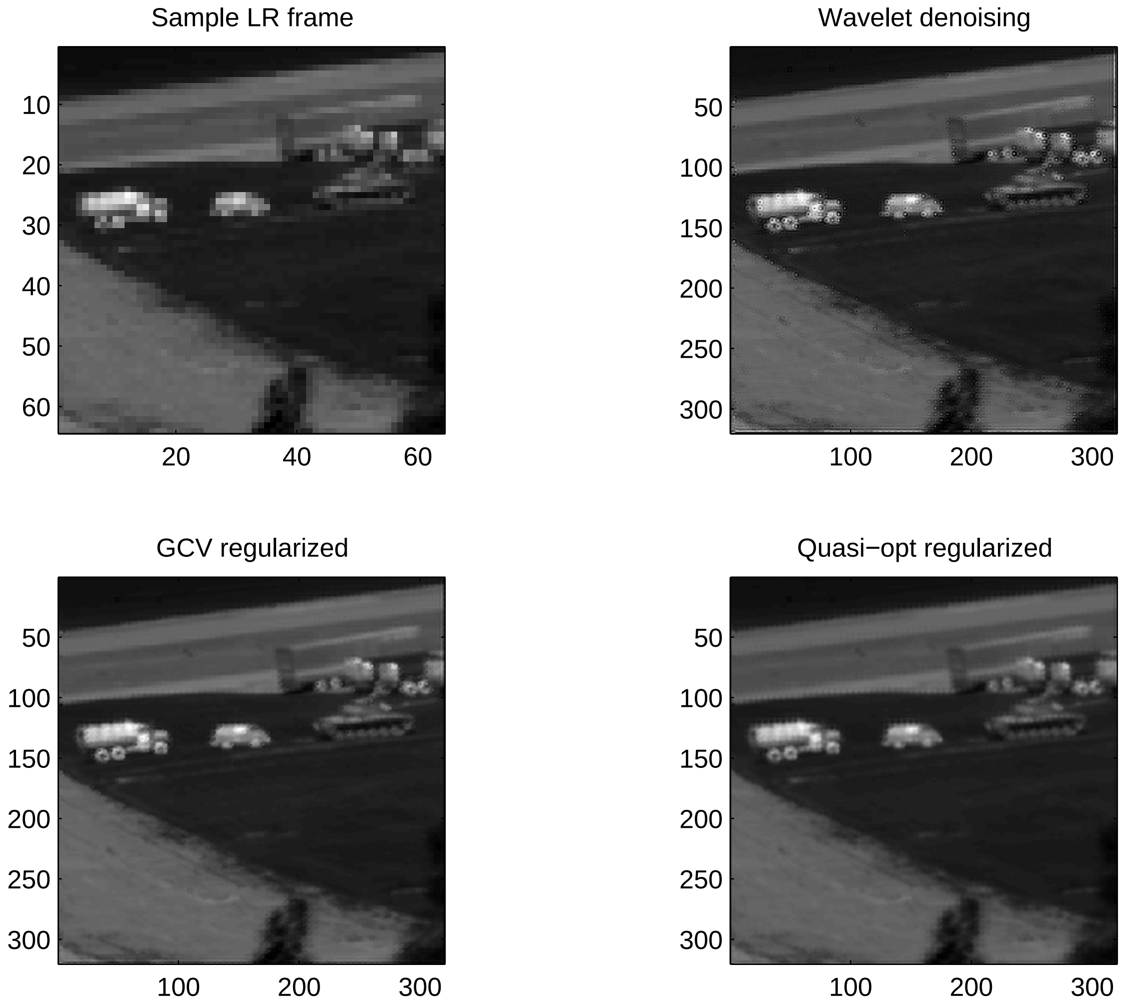


Figure 4.2: Stanford sequence superresolution reconstructions with wavelet denoising, GCV and quasi-optimal GCV regularization.


Sample LR frame

Wavelet denoising

GCV regularized

Quasi−opt regularized

Figure 4.3: FLIR sequence superresolution reconstructions with wavelet denoising, GCV and quasi-optimal regularization.

	Noise standard deviations					
Motion error	0	2	4	6	8	10
0	186.9	363.0	901.6	1755.7	2891.7	4264.2
	186.9	309.5	430.6	533.3	626.2	714.5
	186.9	288.3	397.6	483.1	559.7	631.7
0.05	252.6	424.6	980.0	1852.5	3013.5	4413.9
	242.4	341.7	451.6	554.9	646.1	735.9
	237.0	302.6	404.5	487.9	565.0	636.8
0.10	555.7	622.9	1309.6	2215.9	3421.9	4880.5
	369.6	368.4	466.9	547.4	627.7	704.5
	336.6	344.0	427.1	492.3	557.0	620.9
0.15	797.6	823.5	1543.5	2436.1	3617.8	5051.6
	458.1	398.9	553.9	623.1	694.8	769.0
	371.5	378.0	447.0	509.2	573.8	638.9

Table 4.3: Stanford image sequence reconstruction MSE for wavelet denoising (top), quasi-optimality criterion (middle), and GCV regularization (bottom).

4.5 Conclusion

Regularization of some form is critical to finding stable solutions for linear ill-posed inverse problems. Compared to square/overdetermined least squares problems, regularization for underdetermined least squares has received very little attention. We examined issues in regularizing underdetermined (superresolution) problems, including transformation to standard forms and derivation for two regularization parameter estimation methods. We derived, for underdetermined systems, a slightly different form of the quasi-optimality criterion and the same expression for GCV. We showed experimental results comparing the two methods on our superresolution reconstruction problem. In general, quasi-optimality criterion is inferior to GCV in regularizing our underdetermined superresolution problem.

Chapter 5

Blind Superresolution

5.1 Introduction

Image restoration and image superresolution are closely related problems. In image restoration, the goal is to reconstruct the original image given a degraded image *at the same resolution scale.* Image superresolution reconstructs a higher resolution, restored image from several aliased, degraded, *low resolution* frames. Restoration problems can be modeled simply as

$$\mathbf{f} = C\mathbf{x} + \mathbf{n}, \tag{5.1}$$

where $\mathbf{f}$ is the observed degraded image, $\mathbf{x}$ is the original image we wish to estimate, $\mathbf{n}$ consists of various additive system noise sources, and C is the convolution operator representing the blur. In most instances, blurring is assumed to be linear shift invariant. We recover the original image $\mathbf{x}$ by deconvolving the blur C from the degraded observed $\mathbf{f}$. This classical problem has been thoroughly studied and can be solved by several well-known techniques such as Wiener filtering, recursive Kalman filtering, and iterative deconvolution methods (cf. [4, 92, 10]).

We described in Chapter 1 the connection between image superresolution and restoration. Superresolution includes restoration as a special case. The restoration equation (5.1) can be rewritten within the superresolution framework as

$$\mathbf{f}_k = DCE_k\mathbf{x} + \mathbf{n}_k, \quad 1 \leq k \leq r^2, \tag{5.2}$$

where r is an arbitrary decimation/enhancement factor, the artificial LR "frames" are generated from the degraded image by shifting and down-sampling by a factor of r, and E_k's represent the relative shifts. Clearly, techniques developed for superresolution will be equally applicable for restoration problems.

In many practical applications, the blurring process is not known or known only to within a set of parameters. The problem of restoring the original image from a degraded observation and incomplete information about the blur is called blind deconvolution. Analogously, blind superresolution is superresolving from LR data with incomplete information about the degradation process. There has been extensive work on blind deconvolution. A good survey on the topic can be found in a survey paper by Kundur and Hatzinakos [53]. Existing blind deconvolution methods can be categorized into two main classes: methods which separate blur identification as a disjoint procedure from restoration, and methods which combine blur identification and restoration in one procedure. Methods in the first class tend to be computationally simpler. Blind deconvolution methods can be generalized to handle multiple observations. Multiframe blind deconvolution is better at suppressing noise and edge artifacts and preventing PSF estimates from converging to the trivial delta function.

Intuitively, blind deconvolution is a factorization problem of two-dimensional polynomials in the Z-transform domain. Based on this observation and the fact that multidimensional polynomials are generally not factorizable, Lane and Bates [54] find the blurring and original image polynomial factors by examining the roots of the polynomial of the observed image in the Z-transform domain. Although conceptually attractive, zero sheet separation is highly sensitive to noise. Using multiple LR frames, Shekarforoush and Chellappa [77] proposed estimating the optical transfer function (OTF) by finding spikes in the magnitude of the cross power spectrum of consecutive frames. Several researchers have considered iterative blind restoration. The most popular of these is the iterative blind deconvolution (IBD) method by Ayers and Dainty [6]. IBD simultaneously reconstructs the blur and image values by alternately enforcing constraints in the image and Fourier domain until estimates for both converge. Biggs and Andrews [11] extended the IBD method to multiple frames using the Richardson-Lucy algorithm under a maximum-likelihood (ML) framework. Similar ML approaches were proposed by Sheppard *et al.* [78], Rajagopalan and Chaudhuri [70], Harikumar and Bresler [40] and others. The simulated annealing algorithm [60] also estimates both the blur and image values. This method tries to find a global minimum of a cost function by randomly perturbing blur and image estimates. Kundur and Hatzinakos [53] proposed an iterative approach based on recursive inverse filtering using nonnegativity and support constraints. They used CG to minimize the associated cost function. Several methods identify the blurring process by using special features in the blurred image, such as edges and point sources [65].

Other approaches have simplified the identification problem by parametrizing the PSF. With some knowledge of the imaging system and environment, we can impose a blur degradation model with a few free parameters. Blur identification is then reduced to finding best estimates for these parameters. We generalize this approach to our blind superresolution problem. We enforce a parametric model upon the blurring process. Equation (1.1) becomes

$$\begin{aligned} \mathbf{f}_k &= DC(\sigma)E_k\mathbf{x} + \mathbf{n}_k, \quad 1 \le k \le p && (5.3) \\ &= H_k(\sigma)\mathbf{x} + \mathbf{n}_k. && (5.4) \end{aligned}$$

where the blurring operator C is generated from a parameter set σ. The least squares solution of (5.4) is the minimizer to

$$\text{argmin}_{\mathbf{x}} \sum_{k=1}^{p} \|\mathbf{f}_k - H_k(\sigma)\mathbf{x}\|_2^2 + \lambda \mathbf{x}^T Q \mathbf{x} \tag{5.5}$$

where λ controls the smoothness of the solution, and the stabilization (or regularization) matrix Q is some symmetric positive definite matrix. Without loss of generality, we consider Q to be the identity matrix in this chapter. With known σ, (5.5) is biased toward solutions with low energy. The minimizer to (5.5) can then be expressed as follows:

$$\mathbf{x}(\sigma, \lambda) = (H(\sigma)^T H(\sigma) + \lambda I)^{-1} H(\sigma)^T \mathbf{f} \tag{5.6}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_p \end{bmatrix}, \quad H = \begin{bmatrix} H_1 \\ \vdots \\ H_p \end{bmatrix}, \tag{5.7}$$

where λ is the regularization parameter.

Our approach to blind superresolution first estimates the unknown PSF parameter set σ from raw data. Once an estimate of the PSF is available, the PCG algorithm described in Chapter 3 is used to solve the non-blind superresolution problem. In the next section, we describe our parametrized blur estimation technique using generalized cross-validation. In Section 5.3, we propose a method based on quadrature rules and the Lanczos algorithm, which bounds the GCV criterion value accurately and efficiently. Blur estimation and blind superresolution results are shown in Section 5.6.

5.2 Cross-validation

Generalized cross-validation is a popular method for computing the regularization parameter [30]. In Chapter 4, we derived the formula for GCV for underdetermined linear systems. Not surprisingly, it has the same form as in the overdetermined case:

$$\lambda_{GCV} = \text{argmin}_{\lambda} \frac{\|(HH^T + \lambda I)^{-1}\mathbf{f}\|_2}{\text{tr}((HH^T + \lambda I)^{-1})}. \tag{5.8}$$

Reeves and Mersereau have used GCV for blur identification under an autoregressive moving average (ARMA) model [71]. In a recent study by Chardon, Vozel, and Chehdi [18], GCV has been shown to be an effective tool in parametric blur estimation. Motivated by these successes, we apply GCV to estimate both the PSF and regularization parameters for blind superresolution:

$$\{\sigma_{GCV}, \lambda_{GCV}\} = \text{argmin}_{\{\sigma,\lambda\}} \frac{\|(H(\sigma)H(\sigma)^T + \lambda I)^{-1}\mathbf{f}\|_2}{\text{tr}((H(\sigma)H(\sigma)^T + \lambda I)^{-1})}. \tag{5.9}$$

In [71], Reeves and Mersereau greatly simplified the minimization problem above by assuming the system matrix $H(\sigma)$ to be square and circulant, and hence, diagonalizable by the discrete Fourier transform. As a result, the numerator and denominator of the objective function in (5.9) could be computed easily. However, this approach is not valid for blind superresolution because the system matrix will not typically be square or circulant. Without this circulant assumption, the numerator and denominator of (5.9) are prohibitively expensive to evaluate directly. In the later sections, we describe techniques to bound the objective function (5.9) efficiently and accurately.

Reeves and Mersereau simultaneously estimated the optimal blur and regularization parameters while keeping the image model parameters fixed. We found, however, that by setting the regularization parameter to some small number, the PSF parameters can be better estimated even in the presence of noise and missing frames. We then use the computed PSF to determine the appropriate regularization parameter based upon the data. Our intuition is that with under-regularization, the noise effect is exacerbated and moves the GCV criterion away from possible local minima. Furthermore, the estimated PSF is less biased away from the actual PSF even though the variance of the estimates can be larger. In what follows, we first use a small value of $\lambda = \lambda_0$, so that the estimated PSF parameters can be found by solving a one-dimensional nonlinear optimization problem:

$$\sigma_{GCV} = \mathrm{argmin}_{\sigma} \frac{\|(H(\sigma)H(\sigma)^T + \lambda_0 I)^{-1}\mathbf{f}\|_2}{\mathrm{tr}((H(\sigma)H(\sigma)^T + \lambda_0 I)^{-1})}. \tag{5.10}$$

In the simplest case, the parameter set σ consists of one parameter describing the smoothness of the blur, e.g. the standard deviation of a Gaussian PSF or the radius of a pillbox (out-of-focus) blur. Once a blur estimate $\hat{\sigma}$ is available, we compute the regularization parameter from

$$\hat{\lambda} = \mathrm{argmin}_{\lambda} \frac{\|(H(\hat{\sigma})H(\hat{\sigma})^T + \lambda I)^{-1}\mathbf{f}\|_2}{\mathrm{tr}((H(\hat{\sigma})H(\hat{\sigma})^T + \lambda I)^{-1})}. \tag{5.11}$$

5.3 Quadrature Rules

For large systems, numerators and denominators in (5.10) and (5.11) are very expensive to evaluate directly. We first approximate the denominator using an unbiased trace estimator by Hutchinson [44]. Let U be a discrete random variable that takes the values -1 and $+1$ each with probability $\frac{1}{2}$, and let $\mathbf{u}$ be a vector whose entries are independent samples from U. Then the term $\mathbf{u}^T(HH^T + \lambda I)^{-1}\mathbf{u}$ is an unbiased estimator of $tr((HH^T + \lambda I)^{-1})$.

Now, in order to estimate both the numerators and denominators in (5.10) and

(5.11), we need to estimate quadratic forms $\mathbf{v}^T f(M)\mathbf{v}$, where M is some symmetric positive definite matrix and $f(\xi) = \xi^{-p}$, $p = 1, 2$. There is extensive literature on the application of Gauss quadrature rules to bound bilinear forms; see papers by Golub and collaborators [7, 29, 31, 32, 35]. This chapter applies these techniques for our blur/regularization parameters identification problem.

Let the eigendecomposition of an $n \times n$ matrix M be given by $M = Q^T \Xi Q$, where Q is an orthogonal matrix and Ξ is a diagonal matrix of eigenvalues in increasing order. Then

$$\begin{aligned} \mathbf{v}^T f(M)\mathbf{v} &= \mathbf{v}^T Q^T f(\Xi) Q\mathbf{v} \\ &= \tilde{\mathbf{v}}^T f(\Xi)\tilde{\mathbf{v}} \\ &= \textstyle\sum_{i=1}^n f(\xi_i)\tilde{v}_i^2, \end{aligned} \tag{5.12}$$

where $\tilde{\mathbf{v}} = (\tilde{v}_i) \equiv Q\mathbf{v}$. Suppose that we have bounds on the spectrum of M, e.g. by Gershgorin's circle theorem, $a \le \xi_1 \le ... \le \xi_n \le b$. The last sum can be considered as a Riemann-Stieltjes integral with nondecreasing piecewise constant measure [35]:

$$\sum_{i=1}^n f(\xi_i)\tilde{v}_i^2 = \int_a^b f(\xi)d\mu(\xi), \tag{5.13}$$

where $\mu(\xi)$ is defined as:

$$\mu(\xi) = \begin{cases} 0, & \text{if } \xi < \xi_1 \\ \sum_{j=1}^i \tilde{v}_j^2, & \text{if } \xi_i \le \xi < \xi_{i+1} \\ \sum_{j=1}^n \tilde{v}_j^2, & \text{if } \xi_n \le \xi. \end{cases} \tag{5.14}$$

We can approximate the Riemann-Stieltjes integral (5.13) with Gauss-type quadrature rules. The general form for quadrature rules is

$$\begin{aligned} I[f] &= \sum_{i=1}^k \omega_i f(\theta_i) + \sum_{j=1}^l \nu_j f(\tau_j), \\ \int_a^b f(\xi)d\mu(\xi) &= I[f] + R[f], \end{aligned} \tag{5.15}$$

where the weights ω_i and ν_j and the nodes θ_i are unknown, the nodes τ_j are predetermined, and $R[f]$ is the error term. The Gauss-type quadrature rules differ from one another by the number of prescribed nodes. If there are no prescribed nodes, then we obtain the standard Gauss quadrature:

$$I_G[f] = \sum_{i=1}^{k} \omega_i f(\theta_i). \tag{5.16}$$

If one node is prescribed, we get the Gauss-Radau quadrature rule; with two nodes prescribed, the Gauss-Lobatto rule:

$$I_R[f] = \sum_{i=1}^{k} \omega_i f(\theta_i) + \nu f(\tau), \tag{5.17}$$

$$I_L[f] = \sum_{i=1}^{k} \omega_i f(\theta_i) + \nu_1 f(\tau_1) + \nu_2 f(\tau_2). \tag{5.18}$$

The Gauss-Radau rule is often applied with either $\tau = a$ or $\tau = b$. Gauss-Lobatto has both endpoints prescribed, $\tau_1 = a, \tau_2 = b$. As will be described below, we can compute the unknown nodes θ_i and weights ω_i, ν_i for these Gauss-type rules from recurrence relations of sequences of orthogonal polynomials via the Lanczos bidiagonalization algorithm (cf. [31, 35]). The next subsection describes quadrature bounds on quadratic forms.

5.3.1 Quadrature Error and Bounds

The quadrature error $R[f]$ from (5.15) can be expressed as

$$R[f] = \frac{f^{(2k+l)}(\eta)}{(2k+l)!} \int_a^b \prod_{j=1}^{l} (\xi - \tau_j) \prod_{i=1}^{k} (\xi - \theta_i)^2 d\mu(\xi), \tag{5.19}$$

for some $\eta \in (a, b)$. We have the following bound for the Gauss quadrature rule [31].

Theorem 1 *If* $f^{(2k)}(\xi) > 0,\ \forall k,\ \forall \xi,\ a < \xi < b$, *then*

$$I_G[f] \leq \int_a^b f(\xi) d\mu(\xi). \tag{5.20}$$

Proof. For the Gauss rule, there are no prescribed nodes, so

$$R_G[f] = \frac{f^{(2k)}(\eta)}{(2k)!} \int_a^b \prod_{i=1}^{k} (\xi - \theta_i)^2 d\mu(\xi). \tag{5.21}$$

Since $f^{(2k)}(\eta) > 0,\ \forall k,\ \forall \eta,\ a < \eta < b$, and $\int_a^b \prod_{i=1}^{k} (\xi - \theta_i)^2 d\mu(\xi) \geq 0$, $R_G[f] \geq 0$. Therefore,

$$I_G[f] \leq \int_a^b f(\xi) d\mu(\xi). \tag{5.22}$$

■

Analogous theorems for Gauss-Radau and Gauss-Lobatto rules are presented in the following.

Theorem 2 *If* $f^{(2k+1)}(\xi) < 0,\ \forall k,\ \forall \xi,\ a < \xi < b$, *then*

$$I_{Rb}[f] \leq \int_a^b f(\xi) d\mu(\xi) \leq I_{Ra}[f], \tag{5.23}$$

where I_{Ra} (I_{Rb}) *corresponds to Gauss-Radau rule with prescribed node at* $\tau = a$ (b).

Proof. Similar proof as in Theorem 1.

■

Theorem 3 *If* $f^{(2k)}(\xi) > 0,\ \forall k,\ \forall \xi,\ a < \xi < b$, *then*

$$\int_a^b f(\xi) d\mu(\xi) \leq I_L[f], \tag{5.24}$$

with prescribed nodes $\tau_1 = a$, $\tau_2 = b$.

Proof. Similar proof as in Theorem 1.

■

Recall that the bilinear terms we want to approximate have the form $\mathbf{v}^T M^{-p}\mathbf{v}$, $p = 1, 2$. The function $f(\xi) = \xi^{-p}$ satisfies the hypotheses of Theorems 1–3, for M positive definite ($a > 0$). Hence, we can bound $\int_a^b f(\xi)d\mu(\xi)$ with Gauss-type quadrature rules. Define

$$\begin{aligned} L[f] &:= \max(I_G[f], I_{Rb}[f]), && (5.25) \\ U[f] &:= \min(I_{Ra}[f], I_L[f]), && (5.26) \end{aligned}$$

where I_{Ra} (I_{Rb}) is the Gauss-Radau rule with the prescribed node at $\tau = a$ (b). We have the following bounds [31]:

$$L[f] \leq \int_a^b f(\xi)d\mu(\xi) \leq U[f]. \tag{5.27}$$

As the number of nodes k increases the bounds $L[f]$ and $U[f]$ become tighter. To find quadrature bounds $L[f]$ and $U[f]$, we need the unknown weights ω_i, ν_j, and nodes θ_i. The following section describes how these quantities can be computed from sequences of orthogonal polynomials associated with the weight measure $d\mu(\xi)$.

5.4 Orthogonal Polynomials

For the nondecreasing piecewise constant measure $\mu(\xi)$, we can define a sequence of orthonormal polynomials $\{p_i\}_{i=0}^{n-1}$ such that

$$\int_a^b p_i(\xi)p_j(\xi)d\mu(\xi) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \tag{5.28}$$

These polynomials satisfy a three-term recurrence relation,

$$\xi\mathbf{p}_k(\xi) = T_k\mathbf{p}_k(\xi) + \beta_k p_k(\xi)\mathbf{e}_k, \quad k = 1, ..., n, \tag{5.29}$$

$$p_{-1}(\xi) = 0, \; p_0(\xi) = 1, \tag{5.30}$$

with

$$\mathbf{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{p}_k = \begin{bmatrix} p_0(\xi) \\ \vdots \\ p_{k-1}(\xi) \end{bmatrix}, \quad T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \ddots & \ddots & & \\ & \ddots & \alpha_{k-1} & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix}. \tag{5.31}$$

It can be shown that the nodes θ_i of Gauss quadrature rule are the eigenvalues of T_k, which are also the zeros of the polynomial p_k. The weights ω_i are the square of the first component of the normalized eigenvectors of T_k. The Gauss quadrature approximation is given by (see [34]):

$$I_G\,[f] = \|\mathbf{v}\|^2 \mathbf{u}_1^T f(T_k + \lambda I)\mathbf{u}_1, \tag{5.32}$$

where $\mathbf{u} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T$ is a k-vector with one in the first entry and zeros elsewhere. For the Gauss-Radau rule, we need to adjust the last entry α_k of T_k so that the adjusted tridiagonal matrix $\hat{T}_k$ has an eigenvalue at the prescribed node [29, 7]. For the Gauss-Lobatto rule the last three nonzero entries $\beta_{k-1}, \alpha_k, \beta_{k-1}$ are adjusted to prescribe eigenvalues at a and b. Hence, for Gauss-Radau and Gauss-Lobatto rules, the quadrature approximations have the form:

$$I\,[f] = \|\mathbf{v}\|^2 \mathbf{u}_1^T f(\hat{T}_k + \lambda I)\mathbf{u}_1, \tag{5.33}$$

where $\hat{T}_k$ is the tridiagonal matrix T_k with the last few entries adjusted accordingly.

Following [7], for the Gauss-Radau rule, in order to prescribe a node at either a or b, we replace the last entry α_k of T_k by

$$\phi = \tau + s_{k-1}, \tag{5.34}$$

with $\tau = \{a, b\}$ and s_{k-1} being the last component of the solution $\mathbf{s}$ to

$$\begin{aligned} (T_{k-1} - \tau I)\mathbf{s} &= \beta_{k-1}^2 \mathbf{e}_{k-1}, & (5.35) \\ \beta_{k-1} &= (\gamma_{k-1} + \gamma_k)\delta_{k-1}. & (5.36) \end{aligned}$$

For the Gauss-Lobatto rule, to prescribe nodes at a and b, the last three nonzero entries $\beta_{k-1}, \alpha_k, \beta_{k-1}$ are replaced by ψ, ϕ, ψ, where

$$\phi = \frac{s_{k-1}b - t_{k-1}a}{s_{k-1} - t_{k-1}}, \quad \psi^2 = \frac{b-a}{s_{k-1} - t_{k-1}}, \tag{5.37}$$

where s_{k-1} and t_{k-1} are the last components of the solutions $\mathbf{s}$ and $\mathbf{t}$ to

$$(T_{k-1} - aI)\mathbf{s} = \mathbf{e}_{k-1} \quad (T_{k-1} - bI)\mathbf{t} = \mathbf{e}_{k-1}. \tag{5.38}$$

The entries of T_k itself can be computed via the Lanczos bidiagonalization algorithm as follows.

5.5 Lanczos Bidiagonalization

The matrix T_k is the tridiagonal matrix resulting from k iterations of Lanczos tridiagonalization algorithm for M with $\frac{\mathbf{v}}{\|\mathbf{v}\|_2}$ as the starting vector. Particularly, we are interested in the case where $M = AA^T + \lambda I$. The tridiagonalization of M can be performed more efficiently by applying Lanczos bidiagonalization algorithm to the matrix A. In Table 5.1, we give the algorithm for the Lanczos bidiagonalization factorization to compute the entries of T_k. After k iterations of the Lanczos bidiagonalization algorithm, we get an orthogonal $m \times k$ matrix

$$Q_k = [\mathbf{q}_0 \cdots \mathbf{q}_{k-1}] \tag{5.39}$$

and an $n \times k$ orthogonal matrix

$$W_k = [\mathbf{w}_0 \cdots \mathbf{w}_{k-1}] \tag{5.40}$$

```
$\mathbf{q}_0 = \mathbf{b}/\|\mathbf{b}\|_2$
$\mathbf{s}_0 = A^T\mathbf{q}_0$
$\gamma_1 = \|\mathbf{s}_0\|_2$
$\mathbf{w}_0 = \mathbf{s}_0/\gamma_1$

for $k = 2 : n$ do
    $\mathbf{r}_{k-1} = A\mathbf{w}_{k-2} - \gamma_{k-1}\mathbf{q}_{k-2}$
    $\delta_{k-1} = \|\mathbf{r}_{k-1}\|_2$
    $\mathbf{q}_{k-1} = \mathbf{r}_{k-1}/\delta_{k-1}$
    $\mathbf{s}_{k-1} = A^T\mathbf{q}_{k-1} - \delta_{k-1}\mathbf{w}_{k-2}$
    $\gamma_k = \|\mathbf{s}_{k-1}\|_2$
    $\mathbf{w}_{k-1} = \mathbf{s}_{k-1}/\gamma_k$
end
```

Table 5.1: Lanczos bidiagonalization algorithm.

with the following relations

$$\begin{aligned} AW_k &= Q_kB_k + \delta_k\mathbf{q}_k\mathbf{e}_k^T, \qquad (5.41) \\ A^TQ_k &= W_kB_k^T, \qquad (5.42) \end{aligned}$$

where B_k is a $k \times k$ lower bidiagonal matrix

$$B_k = \begin{bmatrix} \gamma_1 & & & \\ \delta_1 & \ddots & & \\ & \ddots & \ddots & \\ & & \delta_{k-1} & \gamma_k \end{bmatrix} \qquad (5.43)$$

Combining (5.41) and (5.42) we get

$$(AA^T + \lambda I)Q_k = Q_k(B_kB_k^T + \lambda I) + \gamma_k\delta_k\mathbf{q}_k\mathbf{e}_k^T. \qquad (5.44)$$

The tridiagonal T_k is exactly $B_kB_k^T + \lambda I$.

We choose the follow stopping criterion to terminate the Lanczos algorithm:

$$\frac{U\left[f\right]-L\left[f\right]}{U\left[f\right]}\leq 0.01. \tag{5.45}$$

In practice, we use $U\left[f\right]$ as the approximate value for $\int_a^b f(\xi)d\mu(\xi) := \mathbf{v}^T f(M)\mathbf{v}$.

5.6 Experiments

We estimate blur parameters using the GCV criterion with quadrature rules bounds as described above. We use Matlab's CONSTR routine [45] to solve the GCV minimization problem (5.10). For each function evaluation, we iterate with the Lanczos bidiagonalization algorithm until (5.45) is satisfied. For our test image sequences, Lanczos usually terminates within 70 Lanczos iterations, equivalent to 140 matrix-vector multiplies. The iteration count is quite low compared to the dimensions of the system matrix (usually in the tens of thousands).

Example I: In the first set of experiments, 16 LR frames are generated by blurring a 172×172 pixels HR image with a 4×4 Gaussian blur and down-sampling by a factor of 4 in each dimension. We experiment with blurs of standard deviations 0.75, 1.0, and 2.0 and estimate these parameters assuming the support of the PSF to be known. In addition, we consider blind superresolution at 60 dB, 30 dB SNR[1] and without additive Gaussian noise added to the LR frames. We simulate blind superresolution for two cases, with all frames given and with 10 randomly choosen of the 16 available frames. When all frames are available, superresolution is equivalent to a restoration problem. Tables 5.2 and 5.3 display the mean square error (MSE) (see (2.28) for precise definition) (in percent) in the PSF estimates. Figure 5.1 compares the result of blind superresolution, using the computed PSF and regularization parameters from (5.10) and (5.11), against non-blind superresolution. The actual blur standard deviation is 0.75. We add white noise to the LR frames to realize an SNR of 30 dB. The resulting HR estimate is computed from 10 randomly chosen LR frames.

---

[1] Signal to noise ratio (SNR) is defined as $10\log_{10}\frac{\sigma}{\sigma_{\mathbf{n}}}$, where σ, $\sigma_{\mathbf{n}}$ are variances of a clean frame and noise, respectively.

	SNR		
σ	∞ dB	60 dB	30 dB
0.75	3.43	3.27	4.79
1.0	0.02	0.02	0.53
2.0	0.01	0.01	0.11

Table 5.2: % MSE in PSF estimates for Gaussian blur with 10 randomly chosen frames - example I.

Because of accurate PSF estimate, the reconstruction quality for blind and non-blind superresolution is about the same.

In the second set of experiments, we ran tests for a pillbox (defocussed) blur. The parameter to be estimated is the radius of the blur. Our experiments tested for out-of-focus blurs with radii 2 and 5 in Figure 5.2. We plot GCV values for radius taking values from 1 to 10 at 0.2 intervals. The plots show that GCV function achieves global minimum at the correct values in all cases.

Example II: Our final blind superresolution experiment uses the FLIR image sequence, used earlier in Chapter 3. Recall that each image is 64×64 pixels and a resolution enhancement factor of 5 is sought. The objects in the scene are stationary, and 16 frames are acquired by controlled movements of a FLIR imager described in [39]. The frame to frame motions are accurately known for this sequence. We estimate the blur variance assuming a Gaussian blur with support equaling the resolution enhancement factor. Figure 5.3 shows a sample LR frame from the FLIR sequence and the resulting superresolved HR image using the computed variance and blur model. The algorithm can still resolve some high resolution detail even when nothing is known about the blurring process.

5.7 Conclusion

We propose a parametric blur and regularization estimation approach, based on the generalized cross-validation method, for blind restoration/superresolution. We

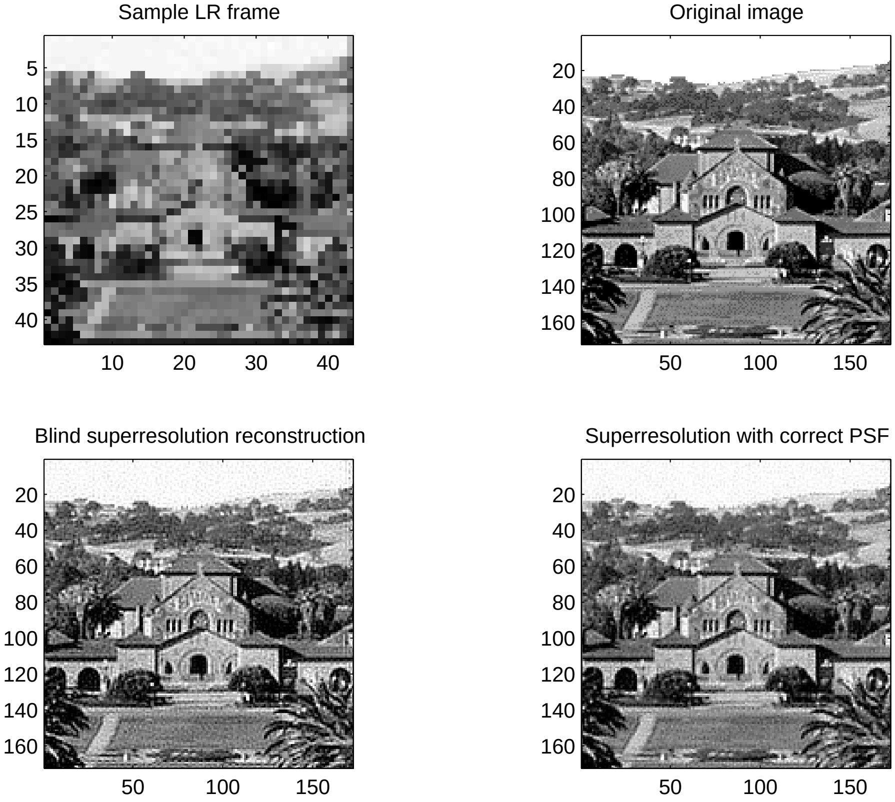


Figure 5.1: Blind superresolution for synthetic sequence - example I.

	SNR		
σ	∞ dB	60 dB	30 dB
0.75	0.00	0.00	2.93
1.0	0.04	0.04	0.17
2.0	0.00	0.00	0.09

Table 5.3: % MSE in PSF estimates for Gaussian blur with all 16 frames available - example I

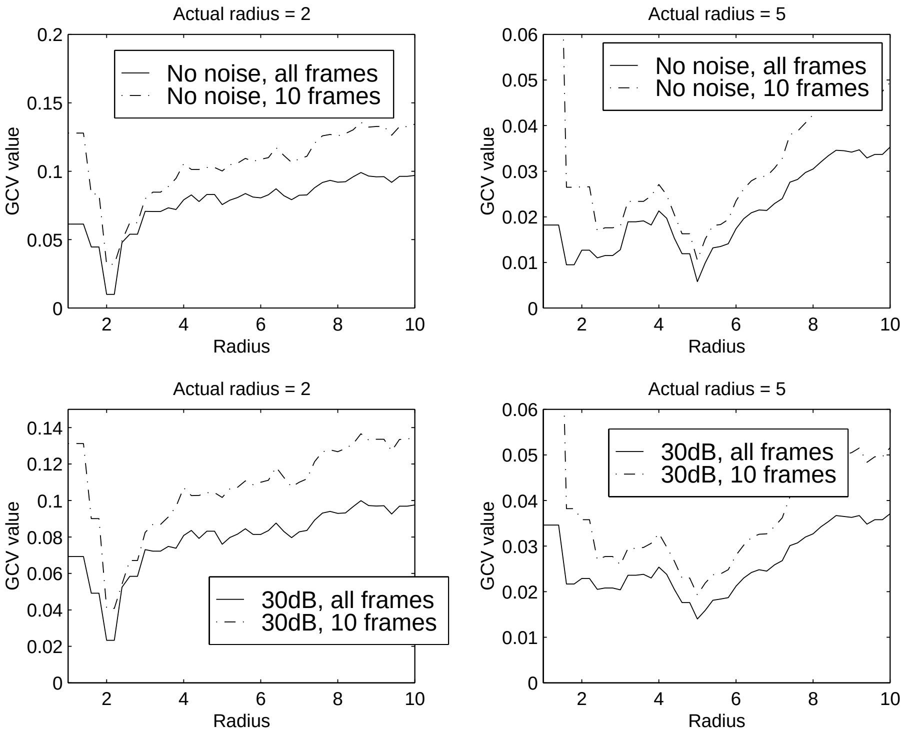


Figure 5.2: GCV plot for pillbox blur - example I.

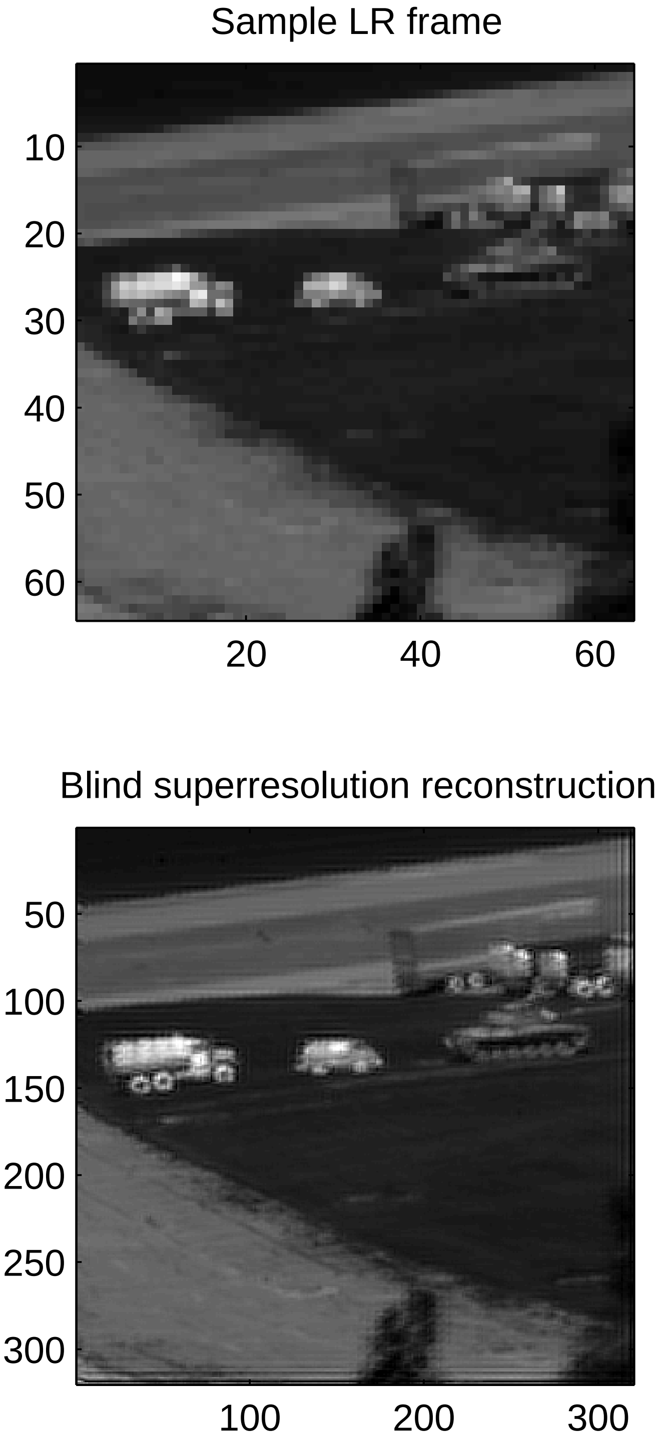


Figure 5.3: Blind superresolution for FLIR sequence - example II.

solve a multivariate nonlinear minimization problem for these unknown parameters. To efficiently and accurately estimate the numerator and denominator of the GCV objective function, we present Gauss-type quadrature techniques for bounding quadratic forms. Experimental results from a synthetic image sequence show that blur parameters are accurately approximated from Gaussian and pillbox blurs under various setup conditions. Image superresolution results using these computed values are visually pleasing as well. Successful experiments with real FLIR image sequence illustrate that our techniques can be a foundation for completely data-driven efficient blind restoration/superresolution algorithms.

Chapter 6

Wavelet Superresolution

6.1 Introduction

This chapter introduces an approach to superresolution completely different from that in Chapters 3 and 4. Instead of viewing superresolution as a total iterative process, we now consider it under the interpolation-restoration framework. As shown in the introduction chapter, the interpolation-restoration approach does exactly what its name suggests: interpolate and then restore values at the HR grid points. We still require motion estimation and blur identification algorithms from Chapters 2 and 5 as preprocessing steps.

We illustrated in Chapter 1 the relationship between superresolution and restoration. We can reduce a superresolution problem to a restoration problem by interpolating values at HR grid points using the LR sample data. Sauer and Allebach [73] were first to consider superresolution as an interpolation problem with nonuniformly sampled data. They used a projection onto convex sets (POCS) algorithm to reconstruct the unknown values. Aizawa *et al.* [3] also modeled superresolution as an interpolation problem with nonuniform sampling and used a formula related to Shannon sampling theorem to estimate values on an HR grid. However, both [3, 73] ignored the effect of sensor blurring. Tekalp *et al.* [83] later extended these algorithms to include blurring and sensor noise and proposed the additional restoration step. Ur and Gross [88] considered Papoulis' generalized multichannel sampling theorem for interpolating values on a higher resolution grid. Because the light detectors are not ideal lowpass filters, some high frequency information about the scene is represented in the image in aliased form. Papoulis' theorem reconstructs this aliased high frequency content by taking a properly weighted sum of the spectral information from the LR frames. Shekarforoush and Chellappa [77] extended Papoulis' theorem for merging of nonuniform samples of multiple channels into HR data.

Reconstructing a HR estimate requires a two-step procedure. First, we interpolate/extrapolate for missing values at the HR grid points in Figure 1.1. Earlier work did not fully exploit the inherent structures and regularity in the data. Namely, if pixel values from all frames are considered together, the data are irregularly sampled. However, for each frame, data points are sampled in a rectangular grid. This special case of irregular sampling is called interlaced sampling [77]. We develop an efficient and elegant wavelet interpolation technique by exploiting this interlacing structure. Having computed these interpolated values, we deblur/restore against the known blurring kernel to obtain an HR estimate.

Previous work did not consider the implications of the 2-D interlaced sampling structure on the computational complexity of the resulting algorithms. The computational complexity for existing algorithms for 2-D data is squared that for 1-D data. As we will show, by exploiting sampling regularity, the computational burden for our algorithm does not drastically increase for 2-D data. In fact, computational complexity of the algorithm for 2-D data is only twice that for 1-D problems. The outline of the rest of the chapter is as follows. We briefly review multiresolution analysis with orthogonal wavelets in Section 6.2. In Section 6.3, we describe our 1-D and 2-D wavelet interpolation method for interlaced data. We discuss computational cost and complexity in Section 6.4. Section 6.5 shows interpolation and superresolution experiments demonstrating the effectiveness of our techniques. We conclude with some comments in Section 6.6.

6.2 Multiresolution Analysis with Orthonormal Wavelets

The fundamental concept behind wavelet theory is the decomposition of signals into components at different scales or resolutions. The advantage of this decomposition is that signal trends at different scales can be isolated and studied. Global trends can be examined at coarser scales while local variations are better analyzed at fine scales. This section presents a brief summary of orthonormal wavelet multiresolution analysis of 1-D and 2-D signals. We briefly review essential ideas necessary for materials in later sections. For more detailed treatments of wavelets, the reader is referred to the excellent books by Strang and Nguyen [82] and Mallat [58].

6.2.1 Multiresolution Analysis for 1-D Signals

Let $L^2(\mathbf{R})$ be the vector space of square-integrable 1-D signals $f(t)$. There exists a sequence of nested approximation subspaces $\mathbf{V}_j, j \in \mathbf{Z}$ and a scaling function $\phi(t)$ satisfying the following requirements [82]:

- $\mathbf{V}_j \subset \mathbf{V}_{j+1}$ and $\bigcap_{j\in\mathbf{Z}} \mathbf{V}_j = \{0\}$ and $\overline{\bigcup_{j\in\mathbf{z}} \mathbf{V}_j} = L^2(\mathbf{R})$.
- $f(t) \in \mathbf{V}_j \Leftrightarrow f(2t) \in \mathbf{V}_{j+1}$.
- $f(t) \in \mathbf{V}_0 \Leftrightarrow f(t-k) \in \mathbf{V}_0$.
- $\mathbf{V}_0$ has an orthonormal basis $\{\phi(t-k)\}$.

Since the set $\{\phi(t-k)\}$ is an orthonormal basis for $\mathbf{V}_0$, dilations and translations of $\phi(t)$, $\{\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)\}_{k\in\mathbf{Z}}$, form an orthonormal basis for $\mathbf{V}_j$. Furthermore, since $\mathbf{V}_0 \subset \mathbf{V}_1$, the scaling function $\phi(t)$ satisfies the following two-scale dilation equation:

$$\phi(t) \quad = \quad \sqrt{2}\sum_k c_k \phi(2t-k), \tag{6.1}$$

for some set of expansion coefficients c_k.

For a function $f(t) \in L^2(\mathbf{R})$, the projection $f_j(t)$ of $f(t)$ onto the subspace $\mathbf{V}_j$ represent an approximation of that function at scale j. The approximation becomes more accurate as j increases. The difference in successive approximations $g_j(t) = f_{j+1}(t) - f_j(t)$ is a detail signal which lives in a wavelet subspace $\mathbf{W}_j$. In fact, we can decompose the approximation space $\mathbf{V}_{j+1}$ as

$$\mathbf{V}_{j+1} = \mathbf{V}_j \oplus \mathbf{W}_j, \tag{6.2}$$

where $\oplus$ denotes the direct sum. Equation (6.2) shows us one of the fundamental reasons why wavelets have been used so successfully in signal processing. Signals can be neatly broken down into a coarse approximation signal and a fine detail signal. Any $f(t) \in L^2(\mathbf{R})$ can be written as a sum of its approximate at some scale J along with the subsequent detail components at scale J and higher. Hence,

$$L^2(\mathbf{R}) \quad = \quad \mathbf{V}_J \oplus \mathbf{W}_J \oplus \cdots \oplus \mathbf{W}_1. \tag{6.3}$$

As in the approximation spaces case, the wavelet spaces $\mathbf{W}_j$ are spanned by a set of orthonormal basis functions $\{\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)\}_{k\in\mathbf{Z}}$, which are dilations and translations of a single wavelet function $\psi(t)$. Furthermore, the wavelet function satisfies the wavelet equation,

$$\psi(t) \quad = \quad \sqrt{2}\sum_k d_k \phi(2t-k), \tag{6.4}$$

for some set of expansion coefficients d_k.

By equation (6.3), we can expand any function $f(t) \in L^2(\mathbf{R})$ as follows:

$$f(t) = \sum_{k\in\mathbf{Z}} a_{J,k}\phi_{J,k}(t) + \sum_{j\geq J}\sum_{k\in\mathbf{Z}} b_{j,k}\psi_{j,k}(t), \tag{6.5}$$

where

$$\begin{aligned} a_{J,k} &= \int f(t)\phi_{J,k}(t)dt \\ b_{j,k} &= \int f(t)\psi_{j,k}(t)dt \end{aligned}$$

are the expansion coefficients for $f(t)$.

6.2.2 Multiresolution Analysis for 2-D Images

The wavelet model in the previous section for 1-D signals can be extended to 2-D images. We describe in this subsection a separable multiresolution analysis of $L^2(\mathbf{R}^2)$ studied by Meyer [62] and Mallat [58]. Given a multiresolution analysis $(\mathbf{V}_j^{(1)})_{j\in\mathbf{Z}}$ of $L^2(\mathbf{R})$, a set of nested subspaces $(\mathbf{V}_j^{(2)})_{j\in\mathbf{Z}}$ forms a multiresolution approximation of $L^2(\mathbf{R}^2)$ with each vector space $\mathbf{V}_j^{(2)}$ being a tensor product of identical 1-D approximation spaces

$$\mathbf{V}_j^{(2)} = \mathbf{V}_j^{(1)} \otimes \mathbf{V}_j^{(1)}. \tag{6.6}$$

Furthermore, the scaling function $\Phi(t, s)$ for the 2-D multiresolution subspaces can be decomposed as:

$$\Phi(t, s) = \phi(t)\phi(s) \tag{6.7}$$

where $\phi(t)$ is the 1-D scaling function of the multiresolution analysis $(\mathbf{V}_j^{(1)})_{j\in\mathbf{Z}}$. The set of functions

$$\Phi_{j,k,l}(t, s) = \phi_{j,k}(t)\phi_{j,l}(s), \quad j, k, l \in \mathbf{Z} \tag{6.8}$$

is an orthonormal basis for $\mathbf{V}_j^{(2)}$. The 2-D wavelet subspaces $\mathbf{W}_j^{(2)}$ are generated by three wavelets to capture detail information in the horizontal, vertical, and diagonal directions

$$\begin{aligned} \Psi^h(t, s) &= \psi(t)\phi(s) & (6.9)\\ \Psi^v(t, s) &= \phi(t)\psi(s) & (6.10)\\ \Psi^d(t, s) &= \psi(t)\psi(s). & (6.11) \end{aligned}$$

The corresponding orthonormal wavelet basis for $\mathbf{W}_j^{(2)}$ is the set

$$\begin{aligned} \Psi^h_{j,k,l}(t, s) &= \psi_{j,k}(t)\phi_{j,l}(s), & (6.12)\\ \Psi^v_{j,k,l}(t, s) &= \phi_{j,k}(t)\psi_{j,l}(s), & (6.13)\\ \Psi^d_{j,k,l}(t, s) &= \psi_{j,k}(t)\psi_{j,l}(s), \quad j, k, l \in \mathbf{Z}. & (6.14) \end{aligned}$$

Analogous to the 1-D case, any image $f(t, s) \in L^2(\mathbf{R}^2)$ can be expanded as a sum of its approximate image at some scale J in $\mathbf{V}_J^{(2)}$ along with subsequent detail components at scale J and higher.

$$\begin{aligned} f(t,s) &= \sum_{k,l\in\mathbf{Z}} a_{J,k,l}\Phi_{J,k,l}(t,s) + \sum_{j\geq J}\sum_{k,l\in\mathbf{Z}} b^h_{j,k,l}\Psi^h_{j,k,l}(t,s) + \\ & \sum_{j\geq J}\sum_{k,l\in\mathbf{Z}} b^v_{j,k,l}\Psi^v_{j,k,l}(t,s) + \sum_{j\geq J}\sum_{k,l\in\mathbf{Z}} b^d_{j,k,l}\Psi^d_{j,k,l}(t,s), \end{aligned} \tag{6.15}$$

with

$$\begin{aligned} a_{J,k,l} &= \int\int f(t,s)\Phi_{J,k,l}(t,s)dtds \\ b^h_{J,k,l} &= \int\int f(t,s)\Psi^h_{J,k,l}(t,s)dtds \\ b^v_{J,k,l} &= \int\int f(t,s)\Psi^v_{J,k,l}(t,s)dtds \\ b^d_{J,k,l} &= \int\int f(t,s)\Psi^d_{J,k,l}(t,s)dtds \end{aligned}$$

The first term on the right hand side of (6.15) represents the coarse scale approximation to $f(t,s)$. The second term represents the detail component in the horizontal direction, the third and fourth the detail components in the vertical and diagonal directions, respectively.

6.3 Wavelet Interpolation of Interlaced Data

This section describes our interpolation technique for interlaced data. We use the expansion formulae (6.5) and (6.15) to first estimate for the wavelet coefficients. Using these estimates, we interpolate for the function values at the HR grid points.

6.3.1 Interpolation for Nonuniformly Sampled 1-D Signals

We first consider the case of nonuniformly sampled 1-D signals. Suppose that we have a function $f(t)$ for which we would like to compute M uniformly distributed values, say, at $t = 0, 1, ..., M-1$. We are given P nonuniformly sampled data points of $f(t)$ at $t = t_0, t_1, ..., t_{P-1}, 0 \leq t_i < M$, where typically, $P < M$.

We take unit-time spacing grid to be resolution level $\mathbf{V}_0$. By repeated application of (6.2), we can decompose $\mathbf{V}_0$ in following fashion:

$$\mathbf{V}_0 \quad = \quad \mathbf{V}_J \oplus \bigoplus_{j=J}^{-1} \mathbf{W}_j, \quad J \leq -1. \tag{6.16}$$

Hence, we can separate $f(t) \in \mathbf{V}_0$ into its approximation and detail components and further expand these components in the orthonormal bases of $\mathbf{V}_J$ and $\{\mathbf{W}_j\}_{-1 \geq j \geq J}$:

$$\begin{aligned} f(t) &= f_J(t) + \sum_{j=J}^{-1} g_j(t), \quad f_J(t) \in \mathbf{V}_J, \; g_j(t) \in \mathbf{W}_j & (6.17) \\ &= \sum_k a_{J,k}\phi_{J,k}(t) + \sum_{j=J}^{-1}\sum_k b_{j,k}\psi_{j,k}(t), & (6.18) \end{aligned}$$

$$\begin{aligned} a_{J,k} &= \int f(t)\phi_{J,k}(t)dt \\ b_{j,k} &= \int f(t)\psi_{j,k}(t)dt. \end{aligned}$$

Substituting in the values of the sampled data we have a set of P linear equations

$$f(t_i) = \sum_k a_{J,k}\phi_{J,k}(t_i) + \sum_{j=J}^{-1}\sum_k b_{j,k}\psi_{j,k}(t_i), \quad i = 0, ..., P-1. \tag{6.19}$$

Suppose that $[0, N]$ is the support interval for $\phi(t)$ and let $t_{max} = \max_i t_i$ and $t_{min} = \min_i t_i$. In the first summation on the right hand side of (6.19), only finitely many terms are nonzero because $\phi_{J,k}(t_i) = \phi(2^J t_i - k)$ is nonzero if and only if $2^J t_i - k$ is in the support interval for $\phi(t)$, i.e.

$$0 \leq 2^J t_i - k \leq N, \tag{6.20}$$

or equivalently,

$$-N + \lceil 2^J t_{min} \rceil \leq k \leq \lfloor 2^J t_{\max} \rfloor. \tag{6.21}$$

Similar arguments can be made for the wavelet basis functions $\psi_{j,k}(t_i)$. Let $S_J = \{-N + \lceil 2^J t_{min} \rceil, ..., \lfloor 2^J t_{\max} \rfloor\}$ be the set of shifts with nonzero contribution in the right hand side of (6.19). We can now rewrite (6.19) as:

$$f(t_i) = \sum_{k \in S_J} a_{J,k} \phi_{J,k}(t_i) + \sum_{j=J}^{-1} \sum_{k \in S_j} b_{j,k} \psi_{j,k}(t_i), \quad i = 0, ..., P-1. \tag{6.22}$$

which, in vector form, becomes

$$\mathbf{f} = G_J \mathbf{a}_J + \sum_{j=J}^{-1} H_j \mathbf{b}_j, \tag{6.23}$$

where

$$\mathbf{f} = (f(t_i))_{i=0,...,P-1}, \ \mathbf{a}_J = (a_{J,k})_{k \in S_J}, \ \mathbf{b}_j = (b_{j,k})_{k \in S_j},$$

$$G_J = (\phi_{J,k}(t_i))_{i=0,...,P-1}^{k \in S_J}, \quad H_j = (\psi_{j,k}(t_i))_{i=0,...,P-1}^{k \in S_j}.$$

To construct G_J and H_j, we need to know basis function values at sampling points $\{t_i\}$. For most wavelet bases, there are no closed-form expressions for basis functions $\{\phi_{J,k}(t), \psi_{j,k}(t)\}$. However, basis function values at dyadic points can be calculated efficiently by recursion. At scale K and for scaling function with support on $[0, N]$, the set of dyadic points is defined to be $\mathcal{D}_K^N = \{0, 1/2^K, ..., N - 1/2^K\}$. We choose K large enough so that the set of sampling points $\{2^J t_i - k, 2^j t_i - k\}$ can be well approximated by a subset of $\mathcal{D}_K^N$.

From Equation (6.23), we can estimate the coarse scale approximation coefficients $\mathbf{a}_J$ by ignoring the detail components and considering just

$$\mathbf{f} \approx G_J \mathbf{a}_J. \tag{6.24}$$

Ford and Etter [27] recommend that J be chosen so that the system above can be solved in an overdetermined least squares sense, that is, $P > \lfloor 2^J t_{max} \rfloor - \lceil 2^J t_{min} \rceil + N + 1$. The scale chosen is dependent on the total number of sample points, the interval spanned by these points, and the support size of the scaling and wavelet functions. Since (6.24) is an approximation, we solve for a *regularized* least squares estimate in the wavelet domain

$$\hat{\mathbf{a}}_J = \text{argmin}_{\mathbf{a}_J} \|\mathbf{f} - G_J \mathbf{a}_J\|_2^2 + \lambda \|\mathbf{a}_J\|_2^2, \tag{6.25}$$

or equivalently,

$$\hat{\mathbf{a}}_J = (G_J^T G_J + \lambda I)^{-1} G_J^T \mathbf{f} \tag{6.26}$$

for some regularization parameter λ. The regularization parameter λ plays a balancing role in (6.25) and (6.26). If λ is too large, the solution obtained will be too far away from the original system we wish to solve. If λ is too small, noise effects are exacerbated in the solution of the under-regularized system. The least squares estimate $\hat{\mathbf{a}}_J$ of the coefficients yields a coarse scale estimate of $\mathbf{f}$, denoted by

$$\hat{\mathbf{f}}_J = G_J \hat{\mathbf{a}}_J, \tag{6.27}$$

The difference between $\mathbf{f}$ and $\hat{\mathbf{f}}_J$ can then be used to approximate the wavelet coeffcients $\mathbf{b}_J$:

$$\begin{aligned} \mathbf{g}_J &= \mathbf{f} - \hat{\mathbf{f}}_J \\ &= \mathbf{f} - G_J \hat{\mathbf{a}}_J \\ &\approx H_J \mathbf{b}_J. \end{aligned} \tag{6.28}$$

Since the number of nonzero coefficients $b_{J,n}$ is the same as the number of nonzero coefficients $a_{J,n}$, equation (6.28) can also be solved in the least squares sense for a regularized estimate of $\mathbf{b}_J$. In general, the regularization parameter is chosen to be small for the coarse scale approximation and large for the fine scale detail because the signal to noise ratio tends to be smaller in the fine scale. In fact, an estimate for λ can be computed using prior information or statistical model for the wavelet coefficients, (cf. [79, 21]). The desired values of $f(t)$ at the HR grid points $t = 0, 1, ..., M-1$ can then be computed using the estimated coefficients:

$$f(t) \approx \sum_{k\in S_J} \hat{a}_{J,k}\phi_{J,k}(t) + \sum_{k\in S_J} \hat{b}_{J,k}\psi_{J,k}(t), \quad t = 0, 1, ..., M-1. \tag{6.29}$$

For wavelet superresolution, the data is sampled nonuniformly but in a recurring manner. This type of sampling is called nonuniform recurring sampling or interlaced sampling [77]. More specifically, we are given sampled data on a LR grid in terms of "frames", which are sets of data points separated by a uniform shift. Let r be the resolution enhancement factor, m be the number of data points per frame, and n be the number of given frames. The available samples are

$$\{f(\epsilon_i), f(r+\epsilon_i), f(2r+\epsilon_i), ..., f((m-1)r+\epsilon_i)\}, \quad 0 \le \epsilon_i < r, \quad i = 1, ..., n.$$

Given these mn sample points, we would like to reconstruct values of $f(t)$ for the HR grid points $t = 0, 1, ..., mr-1$.

Following (6.24) and putting (6.30) in vector form, we get the following set of equations to solve for the coarse scale coefficients $\mathbf{a}_J$

$$\mathbf{f}^{(i)} \approx G_J^{(i)}\mathbf{a}_J, \quad i = 1, ..., n, \tag{6.30}$$

where

$$\mathbf{f}^{(i)} = (f(pr+\epsilon_i))_{p=0,\ldots,m-1}, \quad G_J^{(i)} = (\phi_{J,k}(pr+\epsilon_i))_{p=0,\ldots,m-1}^{k\in S_J}.$$

The vector $\mathbf{f}^{(i)}$ contains sample values from the ith frame. We estimate the wavelet coefficients $\mathbf{b}_J$ and the values of $f(t)$ on the HR grid in the same fashion described previously.

6.3.2 Interpolation for Interlaced 2-D Images

In image superresolution, the data frames are LR rectangular grids of sample points. Let h, w denote the height and width (in units of pixels) of an LR frame. The set of available data is then

$$\{f(pr+\epsilon_{i_t}, qr+\epsilon_{i_s})\},$$
$$0 \leq \epsilon_{i_t}, \quad \epsilon_{i_s} < r, \quad p = 0, ..., h-1, \quad q = 0, ..., w-1, \quad i = 1, ..., n.$$

From these nhw sample points on LR grids, we would like to reconstruct values of $f(t,s)$ on HR grid points $\{(t,s) |\ t = 0, ..., hr-1, s = 0, ..., wr-1\}$. Analogous to the 1-D case, we substitute in sample values of $f(t,s)$ to obtain a set of linear equations and solve a least squares system for the coarse scale coefficients.

$$\begin{aligned} f(pr+\epsilon_{i_t}, qr+\epsilon_{i_s}) &\approx \sum_{k\in S_{J_t}} \sum_{l\in S_{J_s}} a_{J,k,l}\Phi_{J,k,l}((pr+\epsilon_{i_t}, qr+\epsilon_{i_s}) && (6.31) \\ &= \sum_{k\in S_{J_t}} \sum_{l\in S_{J_s}} a_{J,k,l}\phi_{J,k}(pr+\epsilon_{i_t})\phi_{J,l}(qr+\epsilon_{i_s}). && (6.32) \end{aligned}$$

In matrix form, the double sum above can be written as a kronecker product of 1-D wavelet transform matrices

$$\mathbf{f}^{(i)} \approx (G_{J_t}^{(i)} \otimes G_{J_s}^{(i)})\mathbf{a}_J, i = 1, \ldots, n \tag{6.33}$$

where $\mathbf{f}^{(i)}$ is the vector of pixel values of the ith frame reordered rowwise, $\mathbf{a}_J$ is the vector of unknown coarse scale coefficients, and the entries of $G_{J_t}^{(i)}, G_{J_s}^{(i)}$ are basis function values at sampling points of frame i along the horizontal and vertical direction, respectively. Proceeding as in the 1-D case, we solve (6.33) for a regularized least squares estimate $\hat{\mathbf{a}}_J$ of $\mathbf{a}_J$. The difference between $\mathbf{f}^{(i)}$ and its coarse-scale estimate $(G_{J_t}^{(i)} \otimes G_{J_s}^{(i)})\hat{\mathbf{a}}_J$ can next be used to estimate the horizontal detail coefficients

$\mathbf{b}_J^h$:

$$\begin{aligned} \mathbf{g}_J^{(i)} &= \mathbf{f}^{(i)} - (G_{J_t}^{(i)} \otimes G_{J_s}^{(i)})\hat{\mathbf{a}}_J & (6.34) \\ &\approx (G_{J_t}^{(i)} \otimes H_{J_s}^{(i)})\mathbf{b}_J^h. & (6.35) \end{aligned}$$

Continuing as above, the residual is then used to calculate $\mathbf{b}_J^v$ and $\mathbf{b}_J^d$. The choice of scale J makes a crucial difference in reconstruction quality. We pick the finest scale J so that the number of sample values is more than the number of unknown coefficients in (6.33) and (6.35). This gives the finest level of detail coefficients for which we can solve (6.33) and (6.35) stably.

6.4 Implementation and Computational Complexity

We discuss an efficient implementation and the computational complexity of our interpolation approach for interlaced data in this section. We first consider the 1-D interlaced frames equation (6.30). The regularized least squares estimate of the coarse scale coefficients can be expressed as

$$\begin{aligned} \hat{\mathbf{a}}_J &= \left(\left[G_J^{(1)^T} \cdots G_J^{(n)^T} \right] \begin{bmatrix} G_J^{(1)} \\ \vdots \\ G_J^{(n)} \end{bmatrix} + \lambda I \right)^{-1} \left[G_J^{(1)^T} \cdots G_J^{(n)^T} \right] \begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(n)} \end{bmatrix} & (6.36) \\ &= \left(\sum_{i=1}^{n} G_J^{(i)^T} G_J^{(i)} + \lambda I \right)^{-1} \sum_{i=1}^{n} G_J^{(i)^T} \mathbf{f}^{(i)}. & (6.37) \end{aligned}$$

The equation above can be solved most efficiently by an iterative method such as CG (cf. [5]). We only need to compute matrix-vector products involving $\sum_{i=1}^{n} G_J^{(i)^T} G_J^{(i)} + \lambda I$ and not its explicit inverse. Furthermore, because of finite support of $\phi(t)$, the matrices $G_J^{(i)}$ have banded structure which may be further exploited. We can derive similar expressions for the wavelet detail coefficients. For 2-D interlaced images, the regularized least squares estimate for the coarse scale wavelet coefficients from equation (6.33) is

$$\begin{aligned}\hat{\mathbf{a}}_J &= \left(\left[\left(G_{J_t}^{(1)} \otimes G_{J_s}^{(1)} \right)^T \cdots \left(G_{J_t}^{(n)} \otimes G_{J_s}^{(n)} \right)^T \right] \begin{bmatrix} G_{J_t}^{(1)} \otimes G_{J_s}^{(1)} \\ \vdots \\ G_{J_t}^{(n)} \otimes G_{J_s}^{(n)} \end{bmatrix} + \lambda I \right)^{-1} \cdot \\ & \left[\left(G_{J_t}^{(1)} \otimes G_{J_s}^{(1)} \right)^T \cdots \left(G_{J_t}^{(n)} \otimes G_{J_s}^{(n)} \right)^T \right] \begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(n)} \end{bmatrix} \qquad (6.38)\end{aligned}$$

Recalling the following properties of the kronecker product:

- $(A \otimes B)^T = A^T \otimes B^T$,
- $(AB) \otimes (CD) = (A \otimes C)(B \otimes D)$,
- $(A \otimes B)\text{reshape}(V) = \text{reshape}(AVB^T)$ where $\text{reshape}(\cdot)$ reorders the entries of a matrix in rowwise order into vector format,

and applying these properties to the equation above, we have

$$\begin{aligned}\hat{\mathbf{a}}_J &= \left(\sum_{i=1}^{n} \left(G_{J_t}^{(i)T} G_{J_t}^{(i)} \right) \otimes \left(G_{J_s}^{(i)T} G_{J_s}^{(i)} \right) + \lambda I \right)^{-1} \sum_{i=1}^{n} \left(G_{J_t}^{(i)T} \otimes G_{J_s}^{(i)T} \right) \mathbf{f}^{(i)} \qquad (6.39) \\ &= \left(\sum_{i=1}^{n} \left(G_{J_t}^{(i)T} G_{J_t}^{(i)} \right) \otimes \left(G_{J_s}^{(i)T} G_{J_s}^{(i)} \right) + \lambda I \right)^{-1} \sum_{i=1}^{n} \left(G_{J_t}^{(i)T} F^{(i)} G_{J_s}^{(i)} \right) \qquad (6.40)\end{aligned}$$

where $F^{(i)}$ denotes the ith frame in matrix form. Similar forms for wavelet directional detail coefficients estimates can be derived in the same manner. Analogous to the 1-D case, equation (6.40) can be solved most efficiently by an iterative method. Matrix-vector products involving the system matrix $\sum_{i=1}^{n} \left(G_{J_t}^{(i)T} G_{J_t}^{(i)} \right) \otimes \left(G_{J_s}^{(i)T} G_{J_s}^{(i)} \right) + \lambda I$ can exploit computational properties of the kronecker product. We quantify the computational complexity of solving (6.37) and (6.40) in more detail in the following discussion.

The computational burden for the method is comprised of two main components. The first component is the cost of constructing the matrices of wavelet basis functions evaluated at the sampling points. In the 1-D case, at scale K and with wavelet support of size N, the cost for generating the matrices of sampled wavelet basis functions is $\mathcal{O}(N^2 2^{(K+1)})$. Our 2-D wavelet basis is a separable basis, so the construction of the matrices of sampled wavelet basis functions only costs twice as much as in the 1-D case.

The second component of computational burden is the least squares solution of (6.37) and (6.40). We use CG for an iterative solution. Complexity per iteration of CG depends largely on the cost of a matrix-vector product with the system matrix. In the 1-D case the system matrix is

$$\sum_{i=1}^{n} {G_J^{(i)}}^T G_J^{(i)} + \lambda I,$$

with $G_J^{(i)}$, a matrix approximately $m \times \left(2^J mr + N + 1\right)$ in dimensions with m the number of samples per frame. The computational complexity for this matrix-vector product is $\mathcal{O}\left(nm\left(2^J mr + N\right)\right)$. Analogously, the system matrix for interlaced interpolation in 2-D is

$$\sum_{i=1}^{n} \left({G_{J_t}^{(i)}}^T G_{J_t}^{(i)}\right) \otimes \left({G_{J_s}^{(i)}}^T G_{J_s}^{(i)}\right) + \lambda I,$$

where $G_{J_t}^{(i)}$ and $G_{J_s}^{(i)}$ are approximately $h \times \left(2^J hr + N + 1\right)$ and $w \times \left(2^J wr + N + 1\right)$ in dimensions, respectively. The variables h and w denote the height and width of an LR frame in units of samples or pixels. The computational complexity for a matrix vector product with the system matrix is $\mathcal{O}\left(nh\left(2^J hr + N\right) + nw\left(2^J wr + N\right)\right)$. By taking advantage of the interlacing structure and the kronecker product representation, the computational cost for our interpolation approach only doubles for the 2-D case as compared to the 1-D case.

6.5 Experiments

This section presents numerical results for interpolation experiments with 1-D signals and superresolution experiments for 2-D images.

6.5.1 Wavelet Interpolation Experiments for 1-D Signals

In this first set of experiments, we use the wavelet techniques described above to interpolate values of a 1-D signal. We start with an original signal of length 168. The signal is then blurred with a Gaussian point spread function with 3×3 pixel support and variance 1 and down-sampled by a factor of 3 to generate 3 LR frames, each with 56 sample points. We keep only one of those LR frames, leading to a severely underdetermined interpolation problem. The given frame has sample values of $f(t)$ at $t = 0, 3, ..., 165$. The algorithm attempts to reconstruct the signal at time $t = 0, 1, ..., 167$. Figure 6.1 displays the result of wavelet reconstruction using Daubechies DB6 filter [22].

6.5.2 Wavelet Superresolution Experiments for 2-D Images

The setup for the first 2-D superresolution experiment is similar to the experiment for 1-D. The 172×172 pixels HR image shown in the upper left corner of Figure 6.2 is blurred with a Gaussian point spread function of variance 1 with 4×4 pixel support and down-sampled by a factor of 4 to simulate 16 LR frames. We randomly choose 10 of those LR frames, each of size 43×43 pixels, again leading to a severely undersampled superresolution problem. The desired resolution enhancement factor is 4. The wavelet interpolation-restoration process first interpolates at the HR grid points using the proposed wavelet techniques. An estimate for the original HR image is obtained by deconvolving the interpolated values with the known blur. Figure 6.2 shows the result of wavelet superresolution for our test 2-D sequence using Daubechies DB4 filter interpolation in combination with Tikhonov regularized restoration.

As before, each low resolution FLIR image is 64×64 pixels and a resolution

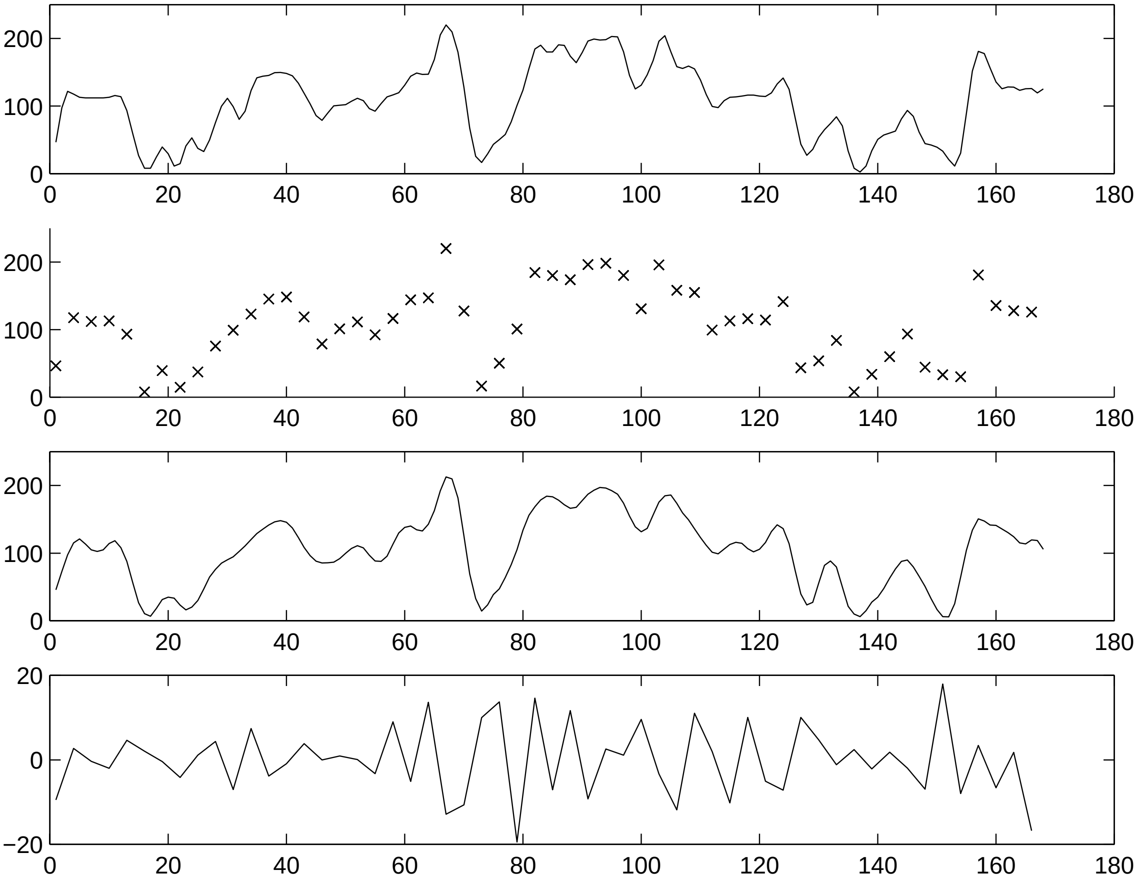


Figure 6.1: Top to bottom. The first plot shows the original signal. The second contains the available data samples. The third plot displays the result of wavelet interpolation at scale $J = -2$ using the available samples. The last graph plots the error of interpolation against the original samples.

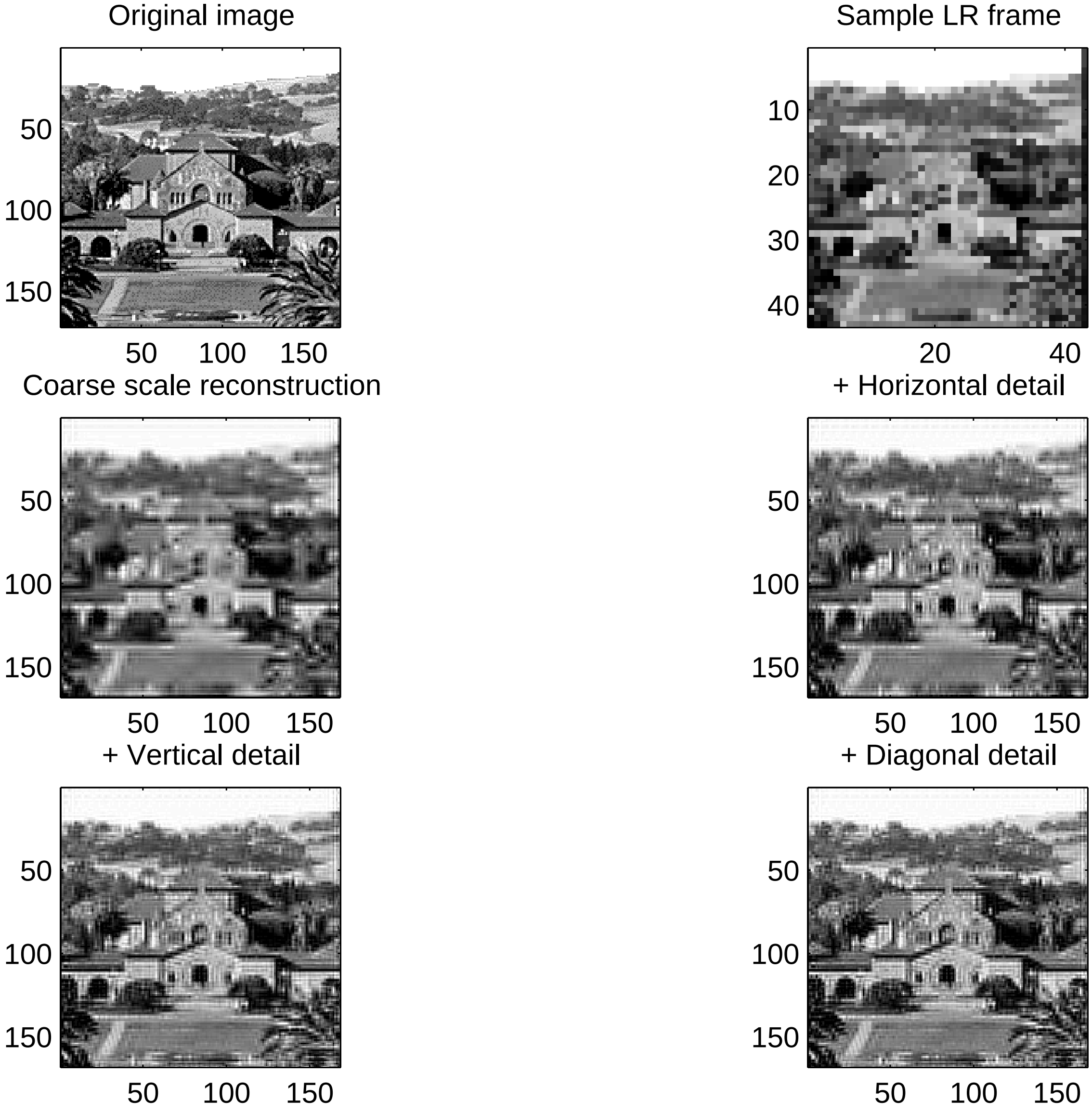


Figure 6.2: The first (upper left) display is the original Stanford HR image. The second (upper right) shows a sample LR frame. The middle left image is the *deblurred* coarse scale approximation. The remaining images are *deblurred* approximations with higher levels of wavelet detail.

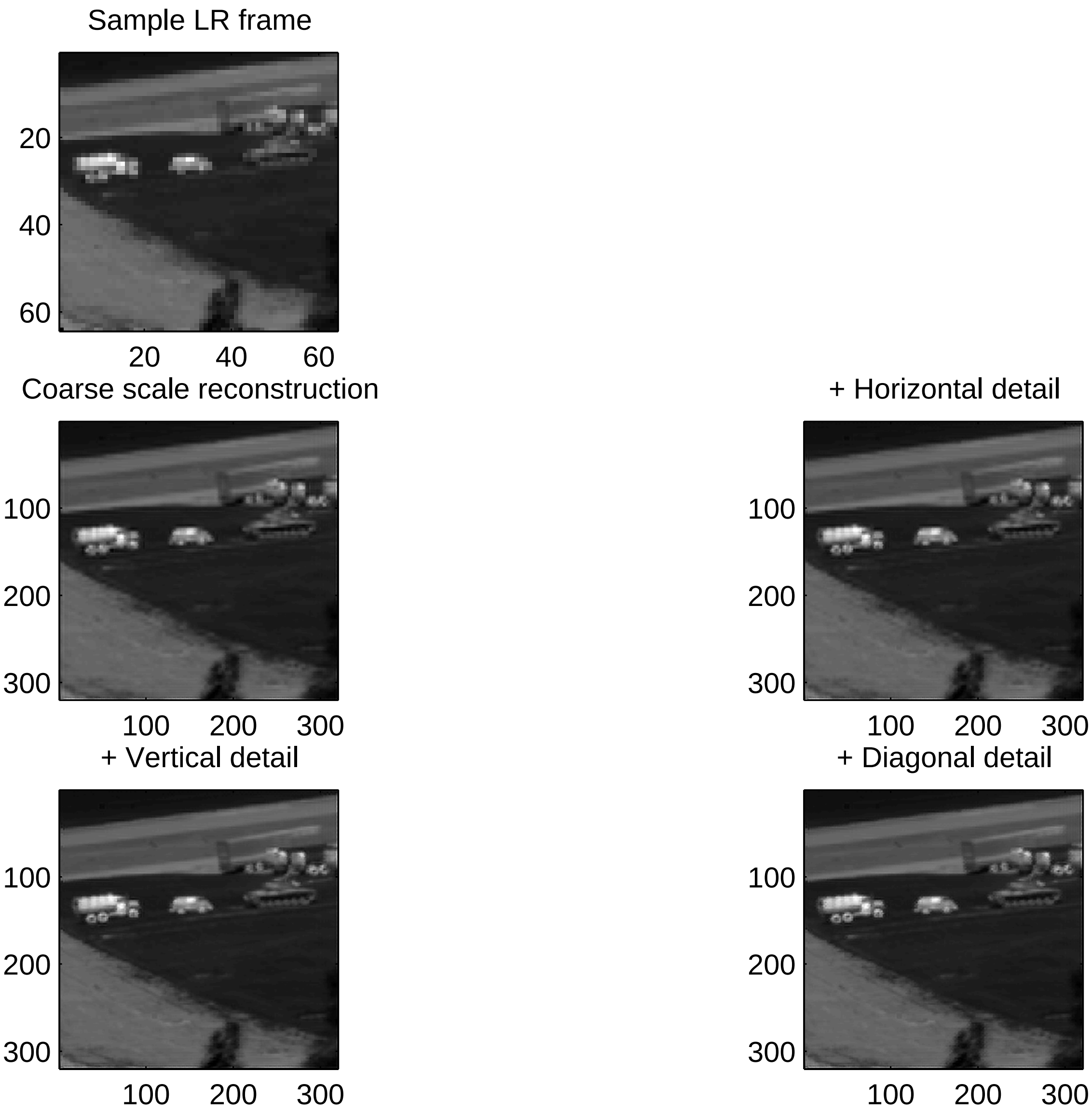


Figure 6.3: The first (upper left) display is a sample FLIR LR frame. The middle left image is the *deblurred* coarse scale approximation. The remaining images are *deblurred* approximations with higher levels of wavelet detail.

enhancement factor of 5 is sought. The objects in the scene are stationary, and 16 frames are acquired by controlled movements of a FLIR imager described in [39]. We applied a similar interpolation-restoration procedure as in the experiment for the simulated Stanford sequence. Figure 6.3 contains the results of wavelet superresolution for the FLIR test sequence using Daubechies DB4 filter interpolation, along with regularized restoration.

6.6 Conclusion

This chapter presents a new wavelet interpolation-restoration method for image superresolution. In contrast to previous interpolation-restoration approaches, our method exploits the interlacing structure of the sampling grid in superresolution. Using a separable orthonormal wavelet basis for 2-D images, we derive a wavelet decomposition using kronecker products. As a result, the computational properties of the kronecker products allow efficient calculation of the wavelet coeffcients. Computational complexity of our method applied to 2-D interlaced data increases only by a factor of 2 compared to that in 1-D. Experimental results with simulated and real FLIR data demonstrate excellent reconstruction quality as well.

Chapter 7

Contributions and Future Work

7.1 Introduction

This last section of the thesis summarizes the primary contributions toward an efficient data-driven superresolution algorithm and some directions for future research. We first outline the novel developments from Chapters 2 through 6 in Section 7.2. Topics for future work are addressed in Section 7.3. We conclude with some final remarks in Section 7.4.

7.2 Contributions

The fundamental objective of our work is the development of efficient and robust data-driven components of a complete superresolution algorithm. Each component of the algorithm must be able to handle the heavy computational burden of processing a large number of data and unknown variables. At the same time, the algorithm itself must be robust with respect to various sources of degradation: noise, blur, quantization, etc. Finally, the entire process must be largely data-driven with minimal human supervision. Very few assumptions about the image characteristics, noise models, or camera models can be made. These stringent constraints allow the most flexibility and adaptibility of the algorithm.

We considered the problem from two complementary directions: iterative approach and interpolation-restoration approach. As described in Chapter 1, both approaches require the same preprocessing steps: measurement of relative motion from frame to frame and blur identification from degraded, LR frames. As Figure 1.5 suggests, the goal is to have computationally inexpensive, self-containing components under a modular framework. Toward this end, Chapter 2 is entirely devoted to the development of motion estimation algorithms, while Chapter 5 mainly concerned itself with the blur identification problem. Chapter 4 considered regularization techniques for obtaining robust solutions for superresolution. Chapter 3 proposed novel preconditioning acceleration for the CG iteration algorithm applied to the regularized least squares superresolution problem. Chapter 6 developed a multiresolution wavelet-based interpolation technique for superresolution interlaced data.

7.2.1 Motion Estimation

Superresolution from multiple frames is only possible in the presence of motion. Because the accuracy of relative frame motion estimates largely determines the quality of superresolution reconstruction, robust and efficient motion estimation techniques are a critical component. Our motion estimation framework relies on the fundamental assumption of intensity conservation. We presented a general affine gradient-based model for motion estimation. We considered the framework above under the Radon transform and derived expressions for affine motion in the projected domain. In particular, 2-D translational motion can be expressed as a combination of two 1-D translational components in the projected domain. This special property of the Radon transform is independent of motion estimation techniques. Hence, we obtain faster 1-D based translation motion estimation methods for their 2-D counterparts. Numerical experiments show that projected estimation techniques are indeed faster than their 2-D counterparts. Complexity analysis further validates this claim. Furthermore, we can bound the degradation in accuracy as a result of the Radon transformation.

The second part of Chapter 2 focused on the effect of misregistration on reconstruction quality. Although image registration is an essential component to superresolution, previous work in the literature has largely ignored its importance. Most have assumed perfect registration or not dealt with misregistration explicitly. We studied superresolution reconstruction mean-squared error under various levels of misregistration and data availability. Our superresolution experimental result yields interesting, but not unexpected, observations: Reconstructed image quality degrades as registration error increases, and with more LR data available, superresolution is more robust with respect to misregistration.

7.2.2 Preconditioning

Superresolution is known to be a computationally very intensive problem. The number of unknown and data variables is on the order of tens of thousands. A practical superresolution algorithm must be efficient in data and unknown calculations. Dynamic superresolution places an even greater computing demand. Dynamic superresolution is superresolved video, where a stream of HR images is produced from a stream of LR frames.

In Chapter 3, we outlined a forward degradation model for LR frames. We applied Krylov subspace methods to solve the Tikhonov-regularized least squares problem based on this degradation model. To accelerate convergence of these subspace methods, CG in particular, we proposed two circulant-type preconditioners based on the degradation model of each LR frame. These preconditioners are applicable in all cases of data availability: underdetermined (fewer frames than r^2, where r is the desired resolution enhancement factor in each dimension), square (r^2 frames), and overdetermined (more than r^2 frames). The proposed block preconditioners exploit the sparse triangular band structure of each block of the system matrix. These preconditioners are shown to be inexpensive to construct, store and solve. Since these block preconditioners are circulant-based, we only need to know and store the first column of each block. Our complexity analysis showed that the cost of solving the preconditioning step is relatively small compared to the cost of one CG iteration. Furthermore, we have convergence proofs for both proposed preconditioners, with bounds on the number of PCG iterations needed to achieve exact convergence. Experimental results with simulated and FLIR image sequence illustrated timing superiority of PCG using the proposed preconditioners over unpreconditioned CG.

7.2.3 Regularization

We require well-behaved superresolution solutions, robust with respect to various sources and levels of noise. In Chapter 4, we reviewed familiar concepts of Tikhonov regularization for stably solving ill-conditioned linear systems. We adopted a general framework for representing regularized least squares problems derived from the forward degradation model. We addressed the issue of regularization parameter estimation for superresolution. The regularization parameter plays an important balancing role in the formulation of a well-conditioned system. Although regularization parameter estimation is a well-studied area of research for overdetermined/square problems, for underdetermined systems, this issue has received relatively little attention in the literature. We extended the derivation for two data-driven regularization parameter estimation techniques to underdetermined systems. We applied the same intuition as in the overdetermined/square case. For the quasi-optimality criterion, our derivation yielded a slightly different formula in the underdetermined case. For the generalized cross-validation method, we obtained the same formula as in the overdetermined case. Numerical experiments show generalized cross-validation to be superior quantitatively and qualitatively to quasi-optimality criterion as a regularization parameter estimation technique. Although the formulae for these parameter estimation techniques were derived in context of the image superresolution reconstruction problem, they may be applicable in other underdetermined ill-posed inverse problems.

7.2.4 Blur Identification and Blind Superresolution

In many practical imaging applications, camera characteristics are unknown or known only up to a few free parameters. The blurring parameters have to be robustly estimated from raw data. In Chapter 5, we proposed the use of generalized cross-validation method for parametrized blur estimation from multiple LR frames. The idea is intuitively appealing: to estimate then validate for the blur parameters that would best fit the given LR data. Previous work has shown that generalized cross-validation provides promising results in blur parameter estimation. However, because the generalized cross-validation objective function is prohibitively expensive to evaluate directly, previous models assumed gross simplifications. In this work, we presented accurate and efficient bounds for the numerator and denominator of the objective function. These bounds are based on moments theory and Gauss quadrature rules. The weights and nodes of these rules are related to the recurrence coefficients of sequences of orthogonal polynomials, which can be generated from the Lanczos bidiagonalization algorithm. In our numerical experiments, the proposed technique provided accurate point spread function parameter estimates for many types of blur and were robust to various noise and data availability conditions. Blind superresolution experiments for simulated and FLIR image sequences demonstrated the feasibility of our technique for practical data-driven blind superresolution applications.

As we illustrated in Chapter 1, restoration can be considered as a special case of superresolution. Techniques proposed for blur identification using multiple LR frames may be used in blind restoration applications as well.

7.2.5 Wavelet Superresolution

Chapter 6 introduced a completely new algorithm for image superresolution. The algorithm belongs to the class of interpolation-restoration methods. As the name suggests, interpolation-restoration methods first interpolate for values on the high resolution grid then deblur to obtain a HR image estimate. Our contribution is a fast, efficient wavelet-based interpolation method that is tailored for superresolution data. As noted in Chapter 1, superresolution data are in interlaced form. Pixels from the same LR frame are periodically sampled in a regular grid. The proposed algorithm exploits this structure and regularity in sampling.

We first reviewed wavelet multiresolution analysis for 1-D signals and 2-D images and described a separable, orthogonal representation with compactly supported wavelet bases for images. Separability combined with the interlacing structure of data samples allows for an efficient and compact representation for images in the wavelet domain using kronecker products. We proposed a robust, multiscale interpolation algorithm based on wavelet coefficients estimation. We presented complexity analysis showing that the computational burden for 2-D interpolation with interlaced data only doubles that of the 1-D case. The accuracy and efficiency of our proposed techniques were illustrated in numerical experiments for 1-D signals and

2-D synthetic and FLIR images.

7.3 Future Work

In this section we outline some possible directions for future research. We start with several generalizations of the static, scalar, single source superresolution problem considered in this thesis: a single HR image output from a sequence of gray scale frames captured with the same imaging system.

The first is the so-called vector (color) superresolution, where each pixel value consists of three color components. The naive approach for this problem would be to consider scalar (gray scale) superresolution for each component separately and merge the HR estimates at the end. This inefficient approach ignores all structure and correlation between components. An interesting research question would be how best to exploit the correlation among color components for a multiframe sequence. Obviously, frame-to-frame motion vectors can be measured more accurately with color vector values. A least squares or cross-validation scheme can be used to obtain the best fitting motion vectors. Similarly, regularization and blur parameters can also be better estimated from multiple color elements. Computationally, we could take advantage of the redundancy of system information among the color components. Since the same motion, regularization and blur parameters are applied for all three components, we only need to solve one equation with multiple right hand sides for the vector superresolution problem.

Vector superresolution can be considered as an extension of superresolution in space while dynamic superresolution is an extension in time. Dynamic superresolution produces a sequence of HR images from a sequence of LR frames. Dynamic superresolution is particularly relevant in generating high resolution video sequences from streams of low resolution video. Again, the question is how best to adapt techniques developed in this thesis in dynamic superresolution applications and what are the key insights to making this problem computationally feasible. There are many times more the amount of data and unknowns in dynamic superresolution than in static superresolution. However, the redundancy and structure in data and problem characteristics can be exploited. We note that in dynamic superresolution, the difference between two consecutive batches of LR frames is just the beginning and ending frames. Thus, motion, regularization and blur parameters may be updated and refined incorporating new frame information and disregarding contributions from the oldest one. Furthermore, the HR estimate from the previous batch is likely to be an excellent initial guess for the current batch. In context of video communications, an extremely useful topic of research that may attract much attention is how to carry the techniques developed here into the compressed data domain. Also, superresolution can be thought of as a compression scheme, where an HR image can be decomposed into several low resolution approximations. The question is how best to pick a subset of LR frames that can faithfully represent the original image.

Yet another possible generalization of superresolution is the multichannel superresolution problem, which combines LR data from different sources. The problem requires separate imaging models for these sources, accounting for different conditions of lighting, camera, motion, blur, noise, etc. The challenge is to extract subpixel information from the given LR data even though they may have undergone different degradation processes. The most promising approach would be a projection onto convex set method, which allows the most flexibility in modeling. The method starts with an initial guess for the desired HR image and sequentially projects and updates the HR estimate using the degradation models as convex constraints. A critical issue is the formation of appropriate and accurate degradation models. Instead of considering the most general multichannel problem, a more tractable and useful subclass would be superresolution for single imaging source under varying lighting, blur, and noise conditions.

The previous paragraphs outlined extensions of static, scalar, single source superresolution to dynamic, vector, and multichannel superresolution. In the following, we list topics of further research within each component of superresolution.

7.3.1 Motion Estimation

Although the motion estimation framework we adopted is found to be sufficient in many superresolution applications, the following is a list of useful extensions and developments for image registration in superresolution:

- Locally adaptive motion estimation - local control of motion vector estimates. Ideally, subpixel information extraction should be done under an optical flow framework allowing for the most general frame-to-frame motion models.
- Accounting for sensor and motion blur - more accurate motion estimation when blur degradations are accounted for. Frame-to-frame relative motion is actual scene motion after undergoing sensor degradations. Only purely translational motion and linear shift invariant blurs commute. Image registration with more general motion or in the presence of shift varying blur must consider blurring effects.
- Iterative refinement - updating and validating motion vector parameters with current HR image estimates. Alternatively updating motion and HR estimates may lead to a more accurate and robust total superresolution scheme.

7.3.2 Preconditioning

Several attractive alternatives to the circulant preconditioners proposed in this thesis deserve attention.

- Other circulant, Toeplitz, or displacement rank preconditioners. The proposed preconditioners are attractive because of the available bounds on their effectiveness. However, other circulant-related preconditioners should be empirically considered as well.
- Multilevel, multigrid preconditioning ideas. Superresolution is inherently multiscale. The concepts and insights of multigrid and multilevel analysis may be very applicable here.

- Extensions of preconditioners to operators. The proposed preconditioners are based on a matrix formulation of purely translational motion along with simple blurring models. More generally applicable preconditioners will consider motion and blur processes as operators. Extensions of preconditioners to Neumann boundary condition are also useful.

7.3.3 Regularization

There are several possible extensions to the Tikhonov regularization framework we presented in Chapter 4. Tikhonov regularization with standard stabilization matrix such as the identity or Laplacian operator under $|| \cdot ||_2$ is an adequate option. However, there is an inherent conflict between standard Tikhonov regularization and superresolution. On one hand, superresolution attempts to extract subpixel and detail information from LR data. On the other hand, Tikhonov regularized reconstruction tends to smooth out noise along with fine detail components.

- Tikhonov regularization with robust estimators - penalty term with total variation or other robust norms. Penalty functionals in Tikhonov regularization, which are more forgiving to image detail and sharp transitions, should be considered.

- Other regularization techniques - maximum a posteriori estimates with edge-preserving prior, regularization with iterative methods. Alternate regularization techniques successful in related image detail reconstruction areas should be considered.

- Automatic regularization parameter estimation - completely data-driven regularization process. Ideally, superresolution is a black-box process with no human intervention. Regularization models as well as parameter estimation must be computed from raw data.

7.3.4 Blur Identification

Blur identification from LR data is a computationally challenging problem. We proposed efficient and accurate blur parameter estimation techniques in this thesis. However, we assumed simple blur models with few free parameters. The following lists possible extensions and avenues of further research.

- Extending the proposed technique to multiple blur parameters with unknown blur model and support. The critical issue is computational feasibility of multiple parameter estimation with very large linear systems.

- Theoretical foundation for nonlinear parameter estimation using generalized cross-validation. Although numerical results have shown generalized cross-validation to be an effective method outside regularization parameter estimation, further analysis is required.

7.3.5 Wavelet Representation

Direct methods are a fast, efficient alternative to their iterative cousins. However, direct methods are derived from compact representation of the given data in indirect domains, e.g. Fourier and wavelets, and not the pixel domain. Hence, we expect reconstruction quality to be inferior to pixel domain iterative techniques.

- Other separable, orthogonal representations for effective interpolation. While the wavelet basis provides a highly compact and efficient representation for images, other bases, which may provide more accurate description for low resolution, blurred, decimated data, should be considered.

- Further error analysis for wavelet superresolution. A better understanding of the relationship between wavelet coefficients reconstruction error and superresolution error is needed.

- Comprehensive study of superresolution approaches, direct versus iterative. Superresolution from multiple frames is a fast growing field of research. Although many methods have been proposed, the field lacks a complete survey and comprehensive comparison between these competing methods. Such work would provide a great service to practitioners.

7.4 Closing

In this thesis, we have developed new techniques and analyses for a complete, efficient, data-driven superresolution algorithm. We have demonstrated their feasibility on simulated and real image sequences. While we believe that the algorithms and numerical experiments presented here can serve as a foundation, much work remains to be done.

Bibliography

[1] I. Abdou. Image restoration of multiple frame sequences. In *Proc. SPIE Applications of Digital Image Processing XIX*, volume 2847, pages 192–202, Denver, CO, Aug. 1996.

[2] H. Aghajan and T. Kailath. SLIDE: Subspace-based line detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:1057–1073, 1994.

[3] K. Aizawa, T. Komatsu, and T. Saito. Acquisition of very high resolution images using stereo cameras. In *Proc. SPIE Visual Communications and Image Processing '91*, volume 1605, pages 318–328, Boston, MA, Nov. 1991.

[4] H. Andrews and B. Hunt. *Digital Image Restoration.* Prentice-Hall, Englewood Cliffs, NJ, 1977.

[5] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, New York, NY, 1994.

[6] G. Ayers and J. Dainty. Iterative blind deconvolution method and its applications. *Optics Letters*, 13(7), Jul. 1988.

[7] Z. Bai, M. Fahey, and G. Golub. Some large-scale matrix computation problems. *J. Comput. Appl. Math.*, 74:71–89, 1996.

[8] J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. *Int. J. Computer Vision*, 12:43–77, 1994.

[9] B. Bascle, A. Blake, and A. Zisserman. Motion deblurring and super-resolution from an image sequence. *EECV*, pages 573–581, Apr. 1996.

[10] J. Biemond, R. Lagendijk, and R. Mersereau. Iterative methods for image deblurring. *Proc. IEEE*, 78(5):856–883, May 1990.

[11] D. Biggs and M. Andrews. Asymmetric iterative blind deconvolution of multiframe images. In F. Luk, editor, *SPIE Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations VII*, volume SPIE Vol. 3461, San Diego, CA, Jul. 1998.

[12] Åke Björck. *Numerical Methods for Least Squares Problems.* SIAM, Philadelphia, PA, 1996.

[13] M. Black. *Robust Incremental Optical Flow.* PhD thesis, Yale University, Sep. 1992.

[14] L. Brown. A survey of image registration techniques. *ACM Computing Survey*, 24(4):325–376, Dec. 1992.

[15] S. Burch, S. Gull, and J. Skilling. Image restoration by a powerful maximum entropy method. *Computer Vision, Graphics, and Image Processing*, 23(2):113–128, Aug. 1983.

[16] R. Chan, T. Chan, M. Ng, W. Tang, and C. Wong. Preconditioned iterative methods for high-resolution image reconstruction from multisensors. In F. T. Luk, editor, *Advanced Signal Processing Algorithms, Architectures, and Implementations VIII*, volume 3461, pages 348–357. SPIE, 1998.

[17] R. Chan and G. Strang. Toeplitz equations by conjugate gradient with circulant preconditioners. *SIAM J. Sci. Stat. Comput.*, 10(1):=104–119, Jan. 1989.

[18] S. Chardon, B. Vozel, and K. Chehdi. A comparative study between parametric blur estimation methods. In *ICASSP '99*, Phoenix, AZ, 1999.

[19] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson. Super-resolved surface reconstruction from multiple images. Technical Report FIA-94-12, NASA, NASA Ames Research Center, Moffett Field, CA, Dec. 1994.

[20] T. Connolly and R. Lane. Gradient methods for superresolution. In *Proc. Int. Conf. on Image Processing*, volume 1, pages 917–920, 1997.

[21] M. Crouse, R. Nowak, and R. Baraniuk. Wavelet-based statistical signal processing using hidden Markov models. *IEEE Trans. on Signal Processing*, 46(4):886–902, Apr. 1998.

[22] I. Daubechies. *Ten Lectures on Wavelets.* SIAM, New York, NY, 1992.

[23] R. Duda and P. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15:11–15, 1972.

[24] M. Elad. *Super-resolution Reconstruction of Images.* PhD thesis, The Technion - Israel Institute of Technology, Dec. 1996.

[25] M. Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE Trans. on Image Processing*, 6(12):1646–1658, 1997.

[26] L. Eldén. A weighted pseudoinverse, generalized singular values, and constrained least squares problems. *BIT*, 22:487–502, 1982.

[27] C. Ford and D. Etter. Wavelet basis reconstruction of nonuniformly sampled data. *IEEE Trans. on Circuits and Systems II*, 45(8):1165–1168, Aug. 1998.

[28] B. Frieden and H. Aumann. Image reconstruction from multiple 1-D scans using filtered localized projection. *Applied Optics*, 26:3615–3621, Sep. 1987.

[29] G. Golub. Some modified matrix eigenvalue problem. *SIAM Review*, 15:318–334, 1973.

[30] G. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21:215–223, 1979.

[31] G. Golub and G. Meurant. Matrices, moments, and quadrature. In D. F. Griffiths and G. A. Watson, editors, *Numerical Analysis 1993*, pages 105–156. Longman, Essex, England, 1994.

[32] G. Golub and Z. Strakos. Estimates in quadratic formulas. *Numer. Algor.*, 8:241–268, 1994.

[33] G. Golub and C. van Loan. *Matrix Computations.* The Johns Hopkins University Press, Baltimore, MD, 2nd edition, 1989.

[34] G. Golub and U. von Matt. Generalized cross-validation for large-scale problems. *Journal of Computational and Graphical Statistics*, 6:1–34, 1997.

[35] G. Golub and J. Welsch. Calculation of Gauss quadrature rules. *Math. Comp.*, 23:221–230, 1969.

[36] E. Haber. *Numerical Strategies for the Solution of Inverse Problems.* PhD thesis, University of British Columbia, Oct. 1997.

[37] M. Hanke and P. Hansen. Regularization methods for large-scale problems. *Surveys on Mathematics for Industry*, 3:253–315, 1993.

[38] M. Hanke and J. Nagy. Toeplitz approximate inverse preconditioner for banded Toeplitz matrices. *Numerical Algorithms*, 7:183–199, 1994.

[39] R. Hardie, K. Barnard, and E. Armstrong. Joint MAP registration and high-resolution image estimation using a sequence of undersampled images. *IEEE Trans. on Image Processing*, 6(12):1621–1633, Dec. 1997.

[40] G. Harikumar and Y. Bresler. Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms. *IEEE Trans. on Image Processing*, 8(2):202–219, Feb. 1999.

[41] D. Heeger. Notes on motion estimation. Class notes - Psych 267/CS 348D/EE 365, Oct. 1996.

[42] G. Holst. *CCD Arrays, Cameras, and Displays.* JCD Publishing, Winter Park, FL, 1996.

[43] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[44] M. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics, Simulation, and Computation*, 19:433–450, 1990.

[45] The Mathworks Inc. MATLAB, high-performance numeric computation and visualization software. Natick, MA, 1992.

[46] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graph. Models Image Processing*, 53:324–335, Dec. 1993.

[47] A. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, Englewood Cliffs, NJ, 1989.

[48] S. Ju, M. Black, and A. Jepson. Skins and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 307–314, San Francisco, CA, Jun. 1996.

[49] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin Internat. Acad. Polon. Sciences et Lettres*, pages 355–357, 1937.

[50] D. Keren, S. Peleg, and R. Brada. Image sequence enhancement using subpixel displacements. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–746, Ann Arbor, MI, Jun. 1988.

[51] S. Kim, N. Bose, and H. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 38(6), Jun. 1990.

[52] S. Kim and W. Su. Recursive high-resolution reconstruction of blurred multiframe images. *IEEE Trans. on Image Processing*, 2(4), Oct. 1993.

[53] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, May 1996.

[54] R. Lane and R. Bates. Automatic multidimensional deconvolution. *J. Opt. Soc. Am. A*, 4(1):180–188, Jan. 1987.

[55] W. Lukosz. Optical systems with resolving power exceeding the classical limit. *J. Opt. Soc. Am.*, 56(11):1463–1472, 1966.

[56] W. Lukosz. Optical systems with resolving power exceeding the classical limit II. *J. Opt. Soc. Am.*, 57(7):932–941, 1967.

[57] S. Mahmoud, M. Afifi, and R. Green. Recognition and velocity computation of large moving objects in images. *IEEE Trans. Acoust. Speech Signal Processing*, 36:1790–1791, 1988.

[58] S. Mallat. A theory for multiresolution in signal decomposition: The wavelet representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(7):674–683, Jul. 1989.

[59] S. Mann and R. Picard. Virtual bellows: Construction high quality stills from video. In *Proc. IEEE Int. Conf. on Image Processing*, Austin, TX, Dec. 1994.

[60] B. McCallum. Blind deconvolution by simulated annealing. *Optics Communications*, 75(2):101–105, Feb. 1983.

[61] P. Mcintosh and G. Veronis. Solving underdetermined tracer inverse problems by spatial smoothing and cross validation. *Journal of Physical Oceanography*, 23:716–730, Apr. 1993.

[62] Y. Meyer. Principe d'incertitude, bases Hilbertiennes et algebres d'operateurs. In *Bourbaki Seminar*, volume 662. Paris, 1985–1986.

[63] P. Milanfar. Projection-based, frequency-domain estimation of superimposed translational motions. *J. Opt. Soc. Am. A*, 13(11), Nov. 1996.

[64] P. Milanfar. A model of the effect of image motion in the Radon transform domain. *IEEE Trans. on Image Proc.*, 8(9):1276–1281, Sep. 1999.

[65] K. Nishi and S. Ando. Blind superresolving image recovery from blur-invariant edges. In *Proc. of ICASSP '94*, volume 5, pages V/85–88, 1994.

[66] C. Paige and M. Saunders. Algorithm 583. LSQR: Sparse linear equations and least squares problems. *ACM Trans. Math. Software*, 8:195–209, 1992.

[67] C. Paige and M. Saunders. LSQR. an algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Software*, 8:43–71, 1992.

[68] A. Papoulis. A new algorithm in spectral analysis and bandlimited extrapolation. *IEEE Trans. Cir. Sys.*, CAS-22(9), Sep. 1975.

[69] A. Patti, M. Sezan, and A. Tekalp. Superresolution video reconstruction with arbitrary sampling lattices and nonzero aperture time. *IEEE Trans. on Image Processing*, 6(8):1064–1076, Aug. 1997.

[70] A. Rajagopalan and S. Chaudhuri. A recursive algorithm for maximum likelihood-based identification of blur from multiple observations. *IEEE Trans. on Image Processing*, 7(7):1075–1079, Jul. 1998.

[71] S. Reeves and R. Mersereau. Blur identification by the method of generalized cross-validation. *IEEE Trans. on Image Processing*, 1(3):301–311, Jul. 1992.

[72] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, Boston, MA, 1996.

[73] K. Sauer and J. Allebach. Iterative reconstruction of band-limited images from non-uniformly spaced samples. *IEEE Trans. Circuits Syst.*, CAS-34:1497–1505, 1987.

[74] M. Saunders. Solution of sparse rectangular systems using LSQR and CRAIG. *BIT*, 35:588–604, 1995.

[75] R. Schultz and R. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE Trans. on Image Processing*, 5:996–1011, Jun. 1996.

[76] N. Shah and A. Zakhor. Resolution enhancement of color video sequences. *IEEE Trans. on Image Processing*, 8(6):879–885, Jun. 1999.

[77] H. Shekarforoush and R. Chellappa. Data-driven multi-channel super-resolution with application to video sequences. *Journal of the Optical Society of America A*, 16(3):481–492, Mar. 1999.

[78] D. Sheppard, B. Hunt, and M. Marcellin. Iterative multiframe superresolution algorithms for atmospheric-turbulence-degraded imagery. *J. Opt. Soc. Am. A*, 15(4):978–992, Apr. 1998.

[79] E. Simoncelli. Statistical models for images: Compression, restoration and synthesis. In *Proc. of the 31st Asilomar Conference on Signals, Systems, and Computers*, pages 673–678, Pacific Grove, CA, Nov. 1997.

[80] D. Slepian and H. Pollak. Prolate spheroidal wave functions, Fourier analysis and uncertainty-I. *BSTJ*, 40:43–62, Jan. 1961.

[81] H. Stark and P. Oskoui. High-resolution image recovery from image-plane arrays, using convex projections. *J. Opt. Soc. Amer. A*, 6:1715–1726, 1989.

[82] G. Strang and T. Nguyen. *Wavelets and Filter Banks.* Wellesley-Cambridge Press, Wellesley, MA, 1997.

[83] A. Tekalp, M. Ozkan, and M. Sezan. High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration. In *Proc. ICASSP '92*, volume 3, pages 169–172, San Francisco, CA, March 1992.

[84] A. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[85] A. Tikhonov and V. Glasko. Use of the regularization method in non-linear problems. *USSR Comput. Math. Math. Phys.*, 5(3):93–107, 1965.

[86] H. Van Trees. *Detection, Estimation and Modulation Theory: Part I.* John Wiley and Sons, New York, NY, 1968.

[87] R. Tsai and T. Huang. Multiframe image restoration and registration. In *Advances in Computer Vision and Image Processing*, volume 1, Greenwich, CT, 1984. JAI.

[88] H. Ur and D. Gross. Improved resolution from subpixel shifted pictures. *CVGIP: Graphical Models and Image Processing*, 54(2):181–186, Mar. 1992.

[89] A. van der Sluis and H. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.

[90] G. Wahba. Practical approximate solutions to linear operator equations when the data are noisy. *SIAM J. Numer. Anal.*, 14:651–667, 1977.

[91] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. on Image Processing*, 3:625–638, 1994.

[92] J. Woods and V. Ingle. Kalman filtering in two dimensions: Further results. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-29:188–197, Apr. 1981.