SUPERRESOLUTION IN IMAGE SEQUENCES

A Thesis Presented

by

Andrey Krokhin

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Master of Science

in

Electrical Engineering

Northeastern University
Boston, Massachusetts

September 2005

NORTHEASTERN UNIVERSITY

Graduate School of Engineering

Thesis Title: Superresolution in Image Sequences

Author: Andrey Krokhin

Department: Electrical and Computer Engineering

Approved for Thesis Requirement of the Master of Science Degree

Dr. Eric Miller, Thesis Advisor   Date

Dr. Edwin Marengo, Thesis Reader   Date

Mr. Michael McCormack, Thesis Reader   Date

Dr. Stephen McKnight, Department Chair   Date

Graduate School Notified of Acceptance:

Dr. Yaman Yener, Director of the Graduate School   Date

NORTHEASTERN UNIVERSITY

Graduate School of Engineering

Thesis Title: Superresolution in Image Sequences

Author: Andrey Krokhin

Department: Electrical and Computer Engineering

Approved for Thesis Requirement of the Master of Science Degree

Dr. Eric Miller, Thesis Advisor Date

Dr. Edwin Marengo, Thesis Reader Date

Mr. Michael McCormack, Thesis Reader Date

Dr. Stephen McKnight, Department Chair Date

Graduate School Notified of Acceptance:

Dr. Yaman Yener, Director of the Graduate School Date

Copy Deposited in Library:

Reference Librarian Date

Table of Contents

Abstract

The term “superresolution” refers to the process of obtaining higher-resolution images from several lower-resolution ones, i.e. resolution enhancement. The quality improvement is caused by fractional-pixel displacements between images. Superresolution allows to overcome the limitations of the imaging system (resolving limit of the sensors) without the need for additional hardware.

This thesis presents a unified matrix-based framework that formulates superresolution as an inverse problem. We show explicitly how to construct every matrix involved in the formulation, and reduce the problem to a single matrix equation. The solution involves matrix inversion via Tikhonov regularization. It is tested on synthetic data, demonstrating the feasibility of the approach, and then results using experimental data are presented. We also investigate the problem of image registration, describing a procedure to measure the shift between images with subpixel accuracy. The results of this work can be used for object tracking and identification.

Acknowledgements

I would like to thank Dr. Eric Miller, my adviser, for his many suggestions and constant support during this research. I am also thankful to the CenSSIS staff, particularly Kristin Hicks, Deanna Beirne, and Anne Magrath for helping with administrative matters.

Funding for the work in this thesis was provided by Textron Systems, Inc. I am thankful for the financial support they have given me. Michael McCormack, who was my contact person for Textron, also generously provided me with some data to experiment with. I would also like to thank him and Prof. Edwin Marengo for agreeing to serve on my thesis committee.

The professors for my graduate courses expanded my knowledge of different aspects of system theory, signal and image processing, pattern recognition, and communications. More importantly, they allowed me to learn to work on more complicated problems than I was used to and understand complex math.

Finally, I am grateful to my family and friends for their emotional support in these sometimes difficult years.

Chapter 1

Introduction and Historical Overview

1.1 Introduction

The goal of superresolution, as its name suggests, is to increase the resolution of an image. Resolution is a measure of frequency content in an image: high-resolution (HR) images are bandlimited to a larger frequency range than low-resolution (LR) images. In cases where information needs to be extracted from images, the more details there are in the image the better. However, the hardware for HR images is expensive and can be hard to obtain. The resolution of digital photographs is limited by the optics of the imaging device. In conventional cameras, for example, the resolution depends on CCD sensor density, which may not be sufficiently high. Infrared and X-ray devices have their own limitations.

Superresolution is an approach that attempts to resolve this problem with software rather than hardware. The concept behind this is time-frequency resolution. Wavelets, filter banks, and the short-time Fourier transform (STFT) all rely on the relationship between time (or space) and frequency and the fact that there is always a tradeoff in resolution between the two.

In the context of superresolution for images, it is assumed that several LR images (e.g. from a video sequence) can be combined into a single HR image: we are decreasing the time resolution, and increasing the spatial frequency content. The LR images cannot all be identical, of course. Rather, there must be some variation between them, such as translational motion parallel to the image plane (most common), some other type of motion (rotation, moving away or toward the camera), or different viewing angles. In theory, the information contained about the object in multiple frames, and the knowledge of transformations between the frames, can enable us to obtain a much better image of the object. In practice, there are certain limitations: it might sometimes be difficult or impossible to deduce the transformation. For example, the image of a cube viewed from a different angle will appear distorted or deformed in shape from the original one, because the camera is projecting a 3-D object onto a plane, and without a priori knowledge of the transformation, it is impossible to tell whether the object was actually deformed. In general, however, superresolution can be broken down into two broad parts: 1) registration of the changes between the LR images, and 2) restoration, or synthesis, of the LR images into a HR image; this is a conceptual classification only, as sometimes the two steps are performed simultaneously.

A huge number of papers has been published on superresolution and related topics since Tsai and Huang's first work in 1984 [2], and it would be impossible to mention every one of them. Here, we try briefly present some the main developments on the topic. Some of the material was adapted from [1], which contains an excellent and detailed overview of history of superresolution; it misses some of the new developments, since the paper was published in 1998. A series of superresolution-related articles that give a good overview of the field also appears in the May 2003 special issue of the *IEEE Signal Processing Magazine.*

1.2 First Formulation

Tsai and Hunag were the first to consider the problem of obtaining a high-quality image from several downsampled and translationally displaced images in 1984 [2]. Their data set consisted of terrestrial photographs taken by Landsat satellites. They modelled the photographs as aliased, translationally displaced versions of a constant scene. Their approach consisted in formulating a set of equations in the frequency domain, by using the shift property of the Fourier transform. Optical blur or noise were not considered. Tekalp, Ozkan and Sezan [3] extended Tsai-Huang formulation by including the point spread function of the imaging system and observation noise.

1.3 Recursive Least Squares

Kim, Bose, and Valenzuela [4] use the same model as Huang and Tsai (frequency domain, global translation), but incorporate noise and blur. Their work proposes a more computationally efficient way to solve the system of equations in the frequency domain in the presence of noise. A recursive least-squares technique is used. However, they do not address motion estimation (the displacements are assumed to be known) or the ill-posedness of the problem due to the presence of zeroes in the PSF. The authors later extended their work to make the model less sensitive to errors by the *total* least squares approach [5], which can be formulated as a constrained minimization problem. This made the solution more robust with respect to uncertainty of motion parameters.

Despite their simplicity and ease of implementation, frequency-domain models have significant drawbacks. They can only accommodate a global translational model, due to the need for an equivalent transformation in the Fourier domain. For the same reason, the noise and degradation models can only be shift-invariant. Finally, since superresoluton is inherently ill-posed, regularization is almost always required. The incorporation of a priori knowledge or constraints is often difficult or inconvenient in the frequency domain. Spatial domain methods, discussed next, address these shortcomings.

1.4 Spatial Domain Methods

Most of the research done on superresolution today is done on spatial domain methods. Their advantages include a great flexibility in the choice of motion model, motion blur and optical blur, and the sampling process. Another important factor is that the constraints are much easier to formulate, for example, Markov random fields or projection onto convex sets (POCS) [1].

1.5 Projection and Interpolation

If we assume ideal sampling by the optical system, then the spatial domain formulation reduces essentially to projection on a HR grid and interpolation of non-uniformly spaced samples (provided motion estimation has already been done). A comparison of HR reconstrucion results with different interpolation techniques can be found in [6] and [7]. Several techniques are given: nearest-neighbor, weighted average, least-squares plane fitting, normalized convolution using a Gaussian kernel, Papoulis-Gerchberg algorithm, and iterative reconstruction. It should be noted, however, that most optical systems cannot be modelled as ideal impulse samplers.

1.6 Probabilistic Methods

Since superresolution involves estimating data or parameters that are unknown, it is natural to model images as probability distribution. Schultz and Stevenson [8] describe discontinuity-preserving prior image model that utilizes Huber Markov Random fields within a Bayesian framework. MAP estimation is done by the gradient projection algorithm, and independent object motion (estimated by hierarchical blocks) is assumed. Motion estimate errors are also modelled in terms of a probability density function. MAP estimation is used in [11] as well, where the problem of segmentation is addressed and rigid-body motion is assumed for more accurate motion estimates. Hardie, Barnard, and Armstrong present a superresolution procedure which is similar to that of Schultz and Stevenson in [16], however they make a significant contribution in [17], where they estimate the HR image and the motion parameters simultaneously. A procedure is suggested where motion and the reconstructed image are estimated alternately, which offers the advantage of not estimating motion directly from LR images. Tom and Katsaggelos [9], on the other hand, use the ML (as opposed to MAP) approach for a degradation model that includes blur and additive noise. Registration and and restoration is performed simultaneously by the expectation maximization (EM) algorithm.

1.7 Iterative Methods

Since superresolution is a computationally intensive process, it makes sense to approach it by starting with a “rough guess” and obtaining successfully more refined estimates. For example, Elad and Feuer [12] use different approximations to the Kalman filter and analyze their performance. In particular, recursive least squares

(RLS), least mean squares (LMS), and steepest descent (SD) are considered. Irani and Peleg [13] describe a straightforward iterative scheme for both image registration and restoration, which uses a back-projection kernel. In their later work [14], the authors modify their method to deal with more complicated motion types, which can include local motion, partial occlusion, and transparency. The basic back-projection approach remains the same, which is not very flexible in terms of incorporating a priori constraints on the solution space. Shah and Zakhor [15] use a reconstruction method similar to that of Irani and Peleg. They also propose a novel approach to motion estimation that considers a set of possible motion vectors for each pixel and eliminate those that are inconsistent with the surrounding pixels.

1.8 Projection Onto Convex Sets (POCS)

In this formulation, constraint sets are defined which limit the solution space for the HR reconstruction. Usually, these sets represent certain desirable characteristics of the image, such as smoothness, positivity, bounded energy, fidelity, etc. The solution is thus reduced to finding the intersection of convex sets. An early work on the subject was done by Stark and Oskoui [18]. They use closedness and convexity of the constraint sets to ensure convergence of iteratively projecting the images onto the sets. However, the solution, in general, is non-unique and dependent on the initial guess. The proposed model does not incorporate noise. Tekalp, Ozkan, and Sezan [20] propose a more robust POCS formulation which incorporates noise, a space-variant PSF of the optical system, and motion blur (due to non-zero aperture time). In general, POCS has the advantages of simplicity, flexibility and generality in the choice of observation model, and ease of inclusion of prior information (which could be defined as another constraint set). However, drawbacks are also present, notably non-uniqueness of solution, dependence on the initial guess, and slow convergence.

1.9 Edge-Preserving Methods

While many works formulate superresolution as a quadratic minimization problem of some kind, Milanfar et al. [19] propose using the L_1 norm both for regularization and for data fusion. This allows better edge preservation, as most quadratic minimization algorithms produce overly smooth images. The total variation (TV) method for denoising and deblurring is proposed, and it is shown that L_1 norm minimization can be implemented as median estimation. The proposed method performs especially well in the presence of non-Gaussian noise (e.g. salt-and-pepper noise), as it eliminates outliers more efficiently.

1.10 Related Topics

There is a variety of topics related to superresolution. For example, motion estimation constitutes a field by itself, used in a range of image and video processing tasks, and it would be impossible to give even a brief overview of related work. However, we will mention a few papers that are related to motion in the context of superresolution, or subpixel motion estimation, which is vital for superresolution. In [21] and [22], a computationally effective method for subpixel image registration (employed specifically for HR restoration), is presented. It is the gradient constraint method, based on Taylor series expansion, and it is used extensively in this thesis. A novel approach to subpixel registration is developed in [23] and [24]. It is the extension of the well-known phase correlation method to subpixel shifts. One technique works in the spatial domain, by extracting the information from secondary peaks in the inverse Fourier transform (IFT) of the phase correlation plot. The other is a frequency-domain method (which does not require IFT). It requires taking the singular value decomposition (SVD) of the phase correlation matrix and fitting a line to the phase component of the dominant singular vectors. Robinson and Milanfar [25] point out that there are theoretical limits to the accuracy of image registration, regardless of the approach used. A few works focus on the computational aspects of superresolution. For example, Nguyen, Milanfar, and Golub [26] propose efficient block circulant preconditioners for solving the Tikhonov-regularized superresolution problem by the conjugate gradient method. Ng and Bose [27] investigate the convergence rate of iterative methods with different preconditioners. An important theoretical result was derived by Lin and Shum [28]. While they do not present any novel HR reconstruction technique, they derive the theoretical and practical performance bounds of superresolution algorithms under various assumption. The results are based on the perturbation theory of linear systems, and show that for large magnification factors, reconstruction-based algorithms are not favorable, and other methods, such as recognition, may be better.

1.11 Thesis Overview

In this thesis, a unified framework was developed that allows to formulate HR image restoration as essentially a matrix inversion, regardless of how it is implemented numerically. Superresolution is treated as an inverse problem, where we assume that LR images are degraded versions of a HR image, even though it may not exist as such. This allows us to put together the building blocks for the degradation model into a single matrix, and the available LR data into a single vector. The formation of LR images becomes a simple matrix-vector multiplication, and the restoration of the HR image a matrix inversion. Constraining of the solution space is accomplished with Tikhonov regularization. The resulting model is intuitively simple (relying on linear algebra concepts) and can be easily implemented in almost any programming environment.

In the next section, we present a quantitative description of the image formation and reconstruction model. In Chapter 3, image registration, which is central to superresolution, is presented. Chapter 4 briefly discusses a different approach to superresoltion, which aims to avoid overly smooth images. Finally, in Chapter 5 we present the practical results obtained by the methods described in this thesis.

Chapter 2

Mathematical Description

2.1 Introduction

In order to apply a superresolution algorithm, a detailed understanding of how images are captured and of the transformations they undergo is necessary. In this section, we develop a model that converts an image that could be obtained with a high-resolution video camera to low-resolution images that are typically captured by a lesser-quality camera. We then attempt to reverse the process to reconstruct the HR image. Our approach is matrix-based. The forward model is viewed as essentially construction of operators and matrix multiplication, and the inverse model as a pseudo-inverse of a matrix.

2.2 Forward model

Let X be a HR grayscale image of size $n_x \times n_y$. Suppose that this image translationally displaced, blurred, and downsampled, in that order. This process is repeated N

times. The displacements may be different each time, but the downsampling factors and the blur remain the same, which is usually true for real-world image acquisition equipment. Let $d_1, d_2, \ldots d_N$ denote the sequence of shifts and r the downsampling factor, which may be different in the vertical and horizontal directions, i.e. there is r_x and r_y. Thus, we obtain N shifted, blurred, decimated versions (observed images) $Y_1, Y_2, \ldots Y_N$ of the original image. Fig. 2.1 shows this process with a sample image.

The "original" image, in the case of real data, may not exist, of course. In that case, it can be thought of as an image that could be obtained with a very high-quality video camera which has a (r_x, r_y) times better resolution and does not have blur, i.e. its PSF is a delta function.

To be able to represent operations on the image as matrix multiplications, it is necessary to convert the image matrix into a vector. Then we can form matrices which operate on each pixel of the image separately. For this purpose, we introduce the operator **vec**, which represents the lexicographic ordering of a matrix. Thus, a vector is formed from vertical concatenation of matrix columns. Let us also define the inverse operator **mat**, which converts a vector into a matrix. To simplify the notation, the dimensions of the matrix are not explicitly specified, but are assumed to be known.

Let $x = \mathbf{vec}(X)$ and $y_i = \mathbf{vec}(Y_i),\ i = 1, \ldots N$ be the vectorized versions of the original image and the observed images, respectively. We can represent the successive

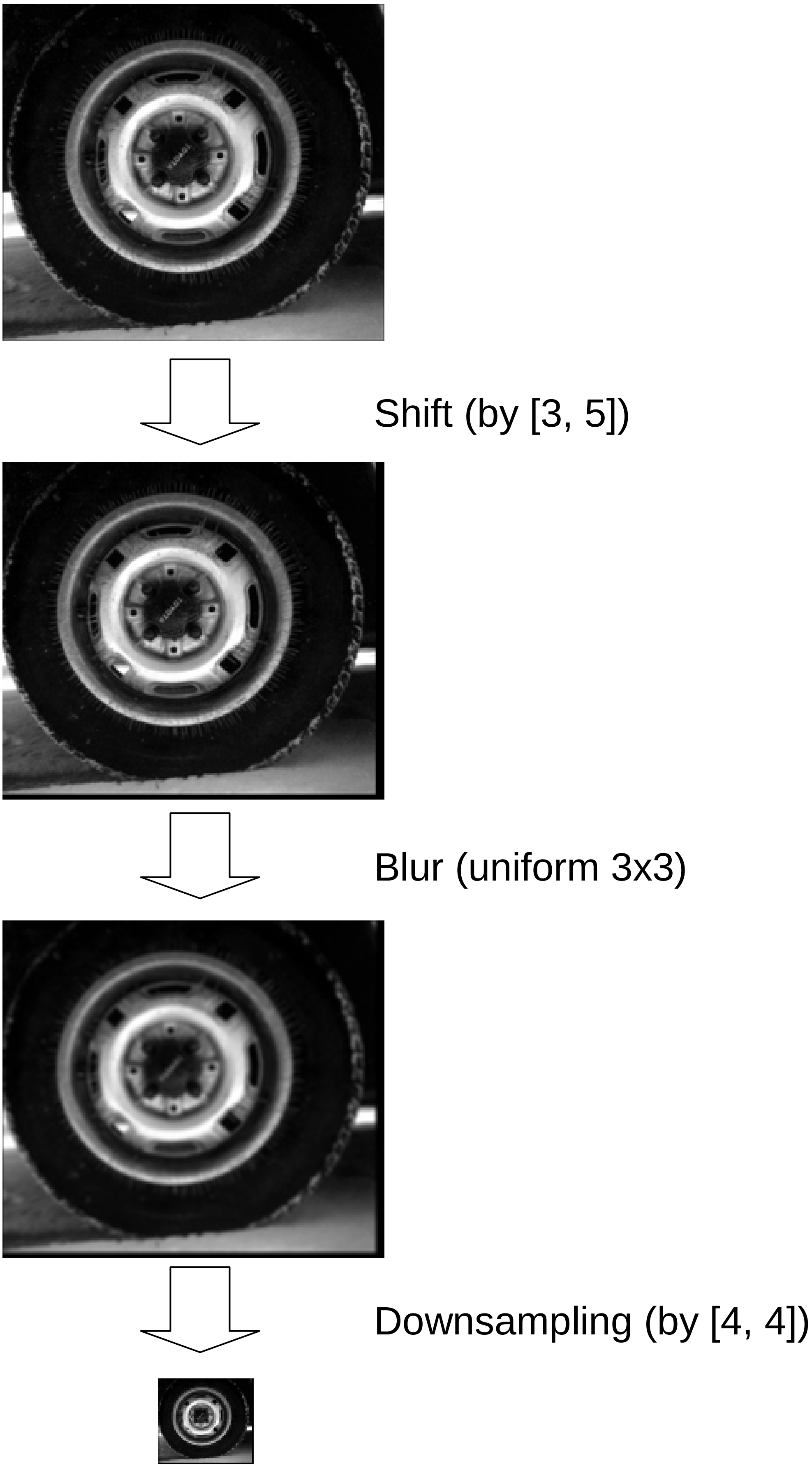


Figure 2.1: Sequence of steps modelling image acquisition.

transformations of x—shifting, blurring, and downsampling—separately from each other.

1) Shift

A shift operator moves all rows or all columns of a matrix up by one or down by one. The row shift operator is denoted by S_x and the columns shift by S_y. Consider a sample matrix

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

After a row shift in the upward direction, this matrix becomes

$$\mathbf{mat}(S_x \mathbf{vec}(M_{ex})) = \begin{bmatrix} 2 & 5 & 8 \\ 3 & 6 & 9 \\ 0 & 0 & 0 \end{bmatrix}$$

Note that the last row of the matrix was replaced by zeros. Actually, this depends on the boundary conditions. In this case, we assume that the matrix is zero-padded around the boundaries, which corresponds to an image on a black background. Other boundary conditions are possible, for example the Dirichlet boundary, when there is no change along the boundaries, i.e. the image's derivative on the boundary is zero. Another case is Neumann boundary condition, where the entries outside the boundary are replicas of those inside. Column shift is defined analogously to the row shift.

Most operators of interest in this thesis have block diagonal form: the only non-zero elements are contained in submatrices along the main diagonal. To represent this, let us use the notation $\mathbf{diag}(A, B, C, \ldots)$ to denote the block-diagonal concatenation of matrices $A, B, C, \ldots$. Furthermore, most operators are composed of the same block repeated multiple times. Let $\mathbf{diag}(\mathbf{rep}(B, n))$ mean that the matrix B is diagonally concatenated with itself n times. Then the row shift operator can be expressed as a matrix whose diagonal blocks consist of the same submatrix B:

$$B = \begin{bmatrix} 0_{(n_x-1)\times 1} & I_{n_x-1} \\ 0_{1\times 1} & 0_{1\times(n_x-1)} \end{bmatrix} \tag{2.2.1}$$

The shift operators have the form:

$$S_x(1) = \mathbf{diag}(\mathbf{rep}(B, n_y)) \tag{2.2.2}$$

$$S_y(1) = \begin{bmatrix} 0_{n_x(n_y-1)\times n_x} & I_{n_x(n_y-1)} \\ 0_{n_x\times n_x} & 0_{n_x\times n_x(n_y-1)} \end{bmatrix} \tag{2.2.3}$$

Here and thereafter, I_n denotes an identity matrix of size n; $0_{n_x\times n_y}$ denotes a zero matrix of size $n_x \times n_y$. The total size of the shift operator is $n_x n_y \times n_x n_y$. The notation $S_x(1)$, $S_y(1)$ simply means that the shift is by one row or column, to differentiate it from the multi-pixel shift to be described later.

As an example, consider a 3×2 matrix M. Its corresponding row shift operators is:

$$S_x(1) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

It is apparent that this shift operator consists of diagonal concatenation of a block B with itself, where

$$B = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

For the column shift operator,

$$S_y(1) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For a shift in the *opposite* direction (the shifts above were assumed to be down and to the right), the operators just have to be transposed. So, $S_x(-1) = S_x^T(1)$, $S_y(-1) = S_y^T(1)$.

Shift operators for multiple-pixel shifts can be obtained by raising the one-pixel shift operator to the power equal to the size of the desired shift. Thus, the notation $S_x(i)$, $S_y(i)$ denotes the shift operator corresponding to the displacement (d_{ix}, d_{iy}) between the frames i and $i-1$, where $S_i = S_x(d_{ix})S_y(d_{iy})$. As an example, consider the shift operators for the same matrix as before, but now for a 2-pixel shift:

$$S_x(2) = S_x^2(1) = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The column shift operator in this case would be an all-zero matrix, since the matrix it is applied to only has two elements itself. However, it is clear how multiple-shift operators can be constructed from single-shift ones. It should be noted that simply raising a matrix to a power may not work for some complicated boundary conditions, such as the reflexive boundary condition. In such a case, the shift operators need to be modified for every shift individually, depending on what the elements outside the boundary are assumed to be.

2) Blur

Blur is a natural property of all image acquisition devices caused by the imperfections of their optical systems. Blurring can also be caused by other factors, such as motion (motion blur) or the presence of air (atmospheric blur), which we do not consider here. Lens blur can be modeled by convolving the image with a mask (matrix) corresponding to the optical system's PSF. Many authors assume that blurring is a simple neighborhood-averaging operation, i.e. the mask consists of identical entries equal to one divided by the size of the mask. Another common blur model is Gaussian. This corresponds to the image being convolved with a two-dimensional Gaussian of size $G_{size} \times G_{size}$ and standard deviation σ^2. Since blurring takes place on the vectorized image, convolution is replaced by matrix multiplication. In general, to represent convolution as multiplication, consider a Toeplitz matrix of the form

$$T = \begin{bmatrix} t_0 & t_{-1} & \dots & t_{2-n} & t_{1-n} \\ t_1 & t_0 & t_{-1} & \ddots & t_{2-n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ t_{n-2} & \ddots & t_1 & t_0 & t_{-1} \\ t_{n-1} & t_{n-2} & \dots & t_1 & t_0 \end{bmatrix} \tag{2.2.4}$$

where negative indices were used for convenience of notation.

Now define the operation $T = \mathbf{toeplitz}(t)$ as converting a vector $t = [t_{1-n}, \dots, t_{-1}, t_0, t_1, t_{n-1}]$ (of length $2n - 1$) to the form (2.2.4), with the negative indices of t corresponding to the first row of T and the positive indices to the first column, with t_0 as the corner element.

Consider a $k_x k_y \times k_x k_y$ matrix T of the form

$$T = \begin{bmatrix} T_0 & T_{-1} & \dots & T_{1-k_y} \\ T_1 & T_0 & T_{-1} & \vdots \\ \vdots & \ddots & \ddots & T_{-1} \\ t_{k_y-1} & \dots & T_1 & T_0 \end{bmatrix} \tag{2.2.5}$$

where each block T_j is a $k_x \times k_x$ Toeplitz matrix. This matrix is called block Toeplitz with Toeplitz blocks (BTTB). Finally, two-dimensional convolution can be converted to an equivalent matrix multiplication form:

$$t * f = \mathbf{mat}(T \ \mathbf{vec}(f)) \tag{2.2.6}$$

where T is the $k_x k_y \times k_x k_y$ BTTB matrix of the form (2.2.5) with $T_j = \mathbf{toeplitz}(t_{\cdot,j})$. Here $t_{\cdot,j}$ denotes the jth column of the $(2k_x - 1) \times (2k_y - 1)$ matrix t [31].

The blur operator is denoted by H. Depending on the image source, the assumption of blur can be omitted in certain cases.

3) Downsampling

The two-dimensional downsampling operator discards some elements of a matrix while leaving others unchanged. In the case of downsampling-by-rows operator, $D_x(r_x)$, the first row and all rows whose numbers are one plus a multiple of r_x are preserved, while all others are removed. Similarly, the downsampling-by-columns operator $D_y(r_y)$ preserves the first column and columns whose numbers are one plus a multiple of r_y, while removing others. As an example, consider the matrix

$$M_{ex} = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 \\ 2 & 6 & 10 & 14 & 18 & 22 & 26 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 \\ 4 & 8 & 12 & 16 & 20 & 24 & 28 \end{bmatrix}$$

Suppose $r_x = 2$. Then we have the downsampled-by-rows matrix

$$\mathbf{mat}(D_x\mathbf{vec}(M_{ex})) = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 & 25 \\ 3 & 7 & 11 & 15 & 19 & 23 & 27 \end{bmatrix}$$

Suppose $r_y = 3$. Then we have the downsampled-by-columns matrix

$$\mathbf{mat}(D_y\mathbf{vec}(M_{ex})) = \begin{bmatrix} 1 & 13 & 25 \\ 2 & 14 & 26 \\ 3 & 15 & 27 \\ 4 & 16 & 28 \end{bmatrix}$$

Matrices can be downsampled by both rows and columns. In the above example,

$$\mathbf{mat}(D_xD_y\mathbf{vec}(M_{ex})) = \begin{bmatrix} 1 & 13 & 25 \\ 3 & 15 & 27 \end{bmatrix}$$

If we define a block matrix B_2,

$$B_2 = \begin{bmatrix} 1 \\ 0_{(r_x-1)\times 1} \end{bmatrix} \tag{2.2.7}$$

then D_x can be written as

$$[\mathbf{diag}(\mathbf{rep}(B_2, n_x n_y/r_x))]^T \tag{2.2.8}$$

For D_y,

$$B_3 = \begin{bmatrix} I_{n_x/r_x} \\ 0_{(n_x(n_y-r_y)/(r_x r_y))\times n_x/r_x} \end{bmatrix} \tag{2.2.9}$$

$$D_y = [\mathbf{diag}(\mathbf{rep}(B_3, n_y/r_y))]^T \tag{2.2.10}$$

It should be noted that the operations of downsampling by rows and columns commute, however, the downsampling operators themselves do not. This is due to the requirement that matrices must be compatible in size for multiplication. If the D_x operator is applied first, its size must be $S_xS_y/r_x \times S_xS_y$. The size of the D_y operator then must be $S_xS_y/(r_xr_y) \times S_xS_y/r_x$. The order of these operators, once constructed, cannot be reversed. Of course, we could choose to construct any operator first.

As an example, consider a 4×4 matrix M_{ex} that is downsampled by 2 in both directions. Its downsampling operators are:

$$D_x = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$D_y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Notice that the downsampling-by-columns operator (D_y) is much smaller that the downsampling-by-rows operator (D_x). This is because D_y will be multiplied not with the original matrix M, but with the smaller matrix $D_x\mathbf{vec}(M_{ex})$, or M_{ex} that has already been downsampled by rows.

Data Model

The observed images are given by:

$$y_i = DHS_i x, \ i = 1, \ldots, N \tag{2.2.11}$$

where $D = D_x D_y$ and $S_i = S_x(d_{ix})S_y(d_{iy})$.

If we define a matrix A_i as the product of downsampling, blurring, and shift matrices,

$$A_i = DHS_i \tag{2.2.12}$$

then the above equation can be written as

$$y_i = A_i x, \ i = 1, \ldots, N \tag{2.2.13}$$

Furthermore, we can obtain all of the observed frames with a single matrix multiplication, rather than N multiplications as above. If all of the vectors y_i are vertically concatenated, the result is a vector y that represents all of the LR frames. Now, the mapping from x to y is also given by the vertical concatenation of all matrices A_i. The resulting matrix A consists of N block matrices, where each block matrix A_i operates the same vector x. By property of block matrices, the product Ax is the same as if all vectors y_i were stacked into a single vector. Hence,

$$y = Ax \tag{2.2.14}$$

The above model assumes that there is a single image that is shifted by different amounts. In practical applications, however, that is not the case. Typically, we are interested in some object that is within the field of view of the video camera. This object is moving while the background remains fixed. If we consider only a few frames (which can be recorded in a fraction of a second), we can define a "bounding box" within which the object will remain for the duration of observation. In this thesis, this "box" is referred to as the *region of interest* (ROI). All operations need to be done only with the ROI, which is much more efficiently computationally. It also poses the additional problem of determining the object's initial location and its movement within the ROI. These issues will be described in the section dealing with motion estimation.

Also, although noise is not explicitly included in the model, the inverse model formulation (described next), assumes that additive white Gaussian (AWGN) noise, if present, can be attenuated by a regularizer, and the degree of attenuation is controlled via the regularization parameter.

2.3 Inverse Model

The goal of the inverse model is to reconstruct a single HR frame given several LR frames. Since in the forward model the HR to LR tranformation is reduced to matrix multiplication, it is logical to formulate the restoration problem as matrix inversion. Indeed, the purpose of vectorizing the image and constructing matrix operators for image transformations was to represent the HR-to-LR mapping in the form of Eq. 2.2.14, a system of linear equations.

First, it should be noted that this system may be underdetermined. Typically, the combination of all available LR frames contains only a part of the information in the HR frame. Alternatively, some frames may contain redundant information (same set of pixels). Hence, straightforward solution of the form $\hat{x} = A^{-1}y$ is not feasible. Instead, we could define the optimal solution as the one minimizing the discrepancy between the observed and the reconstructed data in the least squares sense. For underdetermined systems, we could also define a solution with the minimum norm. However, it is not practical to do so because it is known not known in advance whether the system will be underdetermined. The least-squares solution works in all cases. Let us define a criterion function with respect to $\hat{x}$:

$$J(x) = \lambda ||Qx||_2^2 + ||y - Ax||_2^2 \tag{2.3.1}$$

where Q is the regularizing term and λ its parameter. The solution can then be defined as

$$\hat{x} = \arg\min_{x} J(x) \tag{2.3.2}$$

We can set the derivative of the function to optimize equal to the zero vector and solve the resulting equation:

$$\frac{\partial J(x)}{\partial x} = \mathbf{0} = 2\lambda Q^T Q\hat{x} - 2A^T(y - A\hat{x}) \tag{2.3.3}$$

$$\hat{x} = (A^T A + \lambda Q^T Q)^{-1} A^T y \tag{2.3.4}$$

We can now see the role of the regularizing term. Without it, the solution would have a term $(A^T A)^{-1}$. Multiplication by the downsampling matrix may cause A to have zero rows or zero columns, making it singular. This is intuitively clear, since downsampling is an irreversible operation. The above expression would be non-invertible without the regularizing term, which “fills in” the missing values.

It is reasonable to choose Q to be a derivative-like term. This will ensure smooth transitions between the known points on the HR grid. If we let Δ_x, $Delta_y$ to be the derivative operators, we can write Q as

$$Q = \begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix} \tag{2.3.5}$$

Then

$$Q^T Q = \begin{bmatrix} \Delta_x \\ \Delta_y \end{bmatrix}^T \begin{bmatrix} \Delta_x & \Delta_y \end{bmatrix} = \Delta_x^2 + \Delta_y^2 = L \tag{2.3.6}$$

where L is the discrete Laplacian operator. The Laplacian is a second-derivative term, but for discrete data, it can be approximated by a single convolution with a mask of form

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

The operator L performs this convolution as matrix multiplication. It has the form (2.3.7) below (blanks represent zeroes). For simplicity, this does not take into account the boundary conditions. This should only affect pixels that are on the image's edges, and if they are relevant, the image can be extended by zero-padding.

$$L = \left[\begin{array}{ccccccccccccc}
4 & -1 & 0 & 0 & \ldots & -1 & 0 & 0 & \ldots & & & & 0 \\
-1 & 4 & -1 & 0 & 0 & \ldots & -1 & 0 & 0 & \ldots & & & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & \ldots & -1 & 0 & 0 & \ldots & & 0 \\
 & & & & \ddots & & & & \ddots & & & & \\
-1 & & & -1 & 4 & -1 & & & & -1 & & & \\
0 & -1 & & & -1 & 4 & -1 & & & & -1 & & \\
0 & 0 & -1 & & & -1 & 4 & -1 & & & & -1 & \\
0 & 0 & 0 & \ddots & & & \ddots & \ddots & \ddots & & & & \ddots
\end{array}\right] \tag{2.3.7}$$

The remaining question is how to choose the parameter λ. There exist formal methods for choosing the parameter, such as generalized cross-validation (GCV) or the L-curve, but it is not necessary to use them in all cases: the appropriate value may be selected by trial and error and visual inspection, for example. As noted in [Milanfar], a larger λ makes the system better conditioned, but this new system is farther away from the original system (without regularization). Under the no blur, no noise condition, any sufficiently small value of λ (that makes the matrix numerically invertible) will produce almost the same result. In fact, the difference will probably be lost during round-off, since most grayscale image formats quantize intensity levels to a maximum of 256. When blur is added to the model, however, λ may need to be made much larger, in order to avoid high-frequency oscillations (ringing) in the restored HR image. Since blurring is low-pass filtering, during HR restoration, the inverse process, namely, high-pass filtering, occurs, which greatly amplifies noise. In general, deblurring is an ill-posed problem. Meanwhile, without blurring, restoration is in effect a simple interleaving and interpolation operation, which is not ill-conditioned.

Fig. 2.2 illustrates this. Three HR restoration of the same LR sequences are shown, with different values of the parameter λ. The magnification is by a factor of 2 in both dimensions, and the assumed blur kernel is 3x3 uniform. The image on the left was taken formed with $\lambda = 0.001$, and it is apparent that it is underregularized: noise and motion artefacts have been amplified as a result of deblurring. For the image on the right, $\lambda = 1$ was used. This resulted in an overly smooth image, with few discernible details. The center image is optimal, with $\lambda = 0.11$ as found by GCV. The GCV curve is shown in Fig. 2.3. With deblurring, there is an inevitable tradeoff between image sharpness and the level of noise.

Advantages of the proposed solution

Eq. 2.3.4 produces a vector, which after appropriate reshaping, becomes a HR image. We are interested in how close that restored image resembles the “original”. As mentioned before, in realistic situations the “original” does not exist. The properties

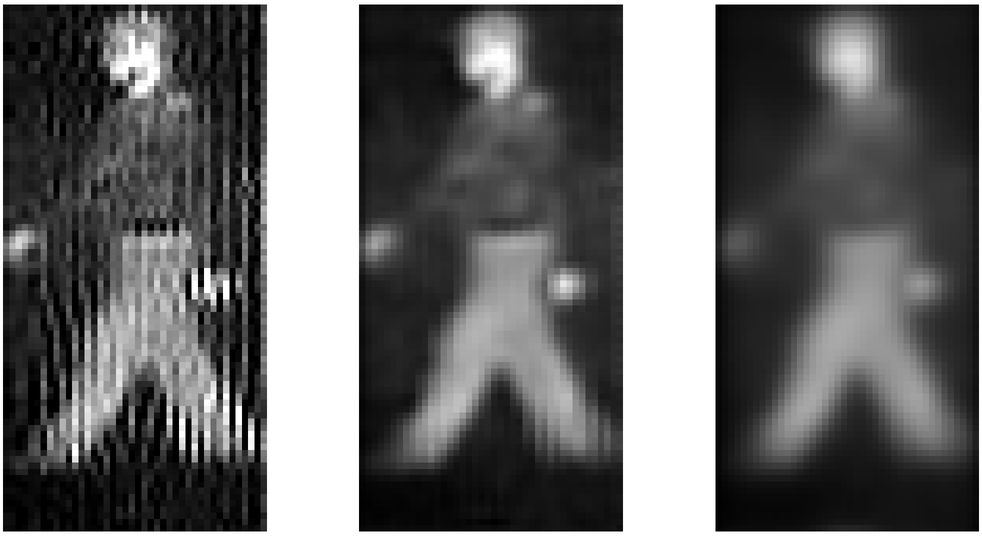


Figure 2.2: Underregularized, optimally regularized, and overregularized HR image.

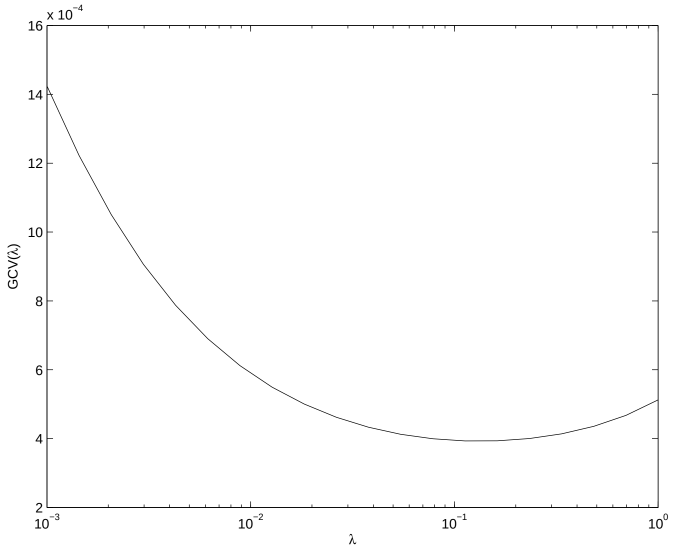


Figure 2.3: Plot of GCV value as a function of λ.

of the solution, however, can be investigated with existing HR images and simulated LR images (formed by shifting, blurring, and downsampling).

Let us define an error metric that formally measures how different the original and the reconstructed HR images are:

$$\varepsilon = \frac{||x - \hat{x}||_2}{||x||_2} \tag{2.3.8}$$

A smaller ε corresponds to a reconstruction that is closer to the original. Clearly, the quality of reconstruction depends on the number of available LR frames and the relative motion between these frames. Suppose, for example, that the downsampling factor in one direction is 4 and the object moves strictly in that direction at 4 HR pixels per frame. Then, in the ideal noiseless case, all frames after the first one will contain the same set of pixels. In fact, each subsequent frames will contain slightly less information, because at each frame some pixels slide past the edge. Now supposed the object's velocity is 2 HR pixels per frame. Than the first two frames will contain unique information, and the rest will be duplicates. The reconstruction obtained with the only the first two frames will be as good as that using many frames.

In the proposed solution, if redundant frames are added, the error as defined by (2.3.8) will stay approximately constant. In the case of real imagery, has the effect of reducing noise due to averaging. Generally speaking, best results are obtained when there are small random movements of the object in both directions (vertically and horizontally). Even if the object remains in place, such movements can obtained by slightly moving the camera.

Under the assumption of no blur and no noise, it can also be shown that there exists a set of LR frames with which almost perfect reconstruction is possible. LR frames can be thought of as being mapped onto the HR grid. If all points on the grid are filled, the image is perfectly reconstructed. Suppose, for example, that the original HR image is downsampled by (2, 3) (2 by rows and 3 by columns)[1]. Suppose the first LR frame is generated by downsampling the HR image with no motion, i.e. its displacement is (0, 0). Then the set of LR frames with the following displacements is sufficient for reconstruction:

$$\begin{gathered}(0,\ 0),\ (0,\ 1),\ (0,\ 2),\\ (1,\ 0),\ (1,\ 1),\ (1,\ 2).\end{gathered}$$

In general, for downsampling by (r_x, r_y), all combinations of shifts from 0 to r_x and 0 to r_y are necessary to fully reconstruct the image. If (2.3.4) is used, the error defined by (2.3.8) will be almost zero. The very small residual is due to the presence of the regularization term and boundary effects.

Figs. 2.4 to 2.7 illustrate the proposed restoration algorithm. The original HR image is an aerial view of a city. The HR-to-LR transformations, with a downsampling factor of (4, 4) and no blur for simplicity, were applied to generate the LR

[1] The convention for representing downsampling is to list it as (downsampling by rows, downsampling by columns). The convention for representing displacements is to list them as (rows, columns), where positive values for rows represent a downward motion and positive values for columns represent a rightward motion; this corresponds to a coordinate system with the origin at the image's upper left corner, with axes pointing down and right.

images. Introducing different displacements between these images, and using a different number of these images, HR reconstructions with different degrees of accuracy were created. In Fig. 2.5, the motion is strictly horizontal with a constant velocity of (0, 3) pixels per frame (all velocities are given in HR pixels). For diagonal motion, Fig. 2.6, the image was displaced by (-1, 3) pixels in each frame. For zigzag motion, Fig. 2.7, a motion pattern was selected that eventually covered all points on the HR grid:

[(-2,-3); (1,2); (-2,-1); (3,-1); (-3,3); (2,-2); (-2,1); (2,-2); (1,2); (-2,-1); (-1,-1); (1,2); (0,1); (1,0); (1,-2)]

Note that a near-perfect reconstruction is possible when a “good” LR sequence is available, even though each of the LR images by itself has 16 times less information than the original.

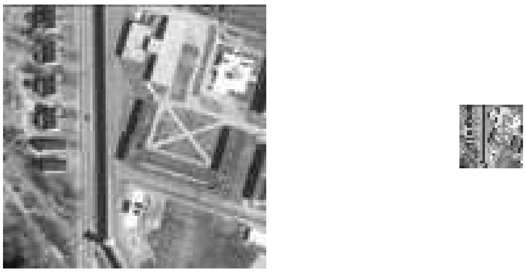


Figure 2.4: The original HR image and a LR image obtained by downsampling by (4, 4).

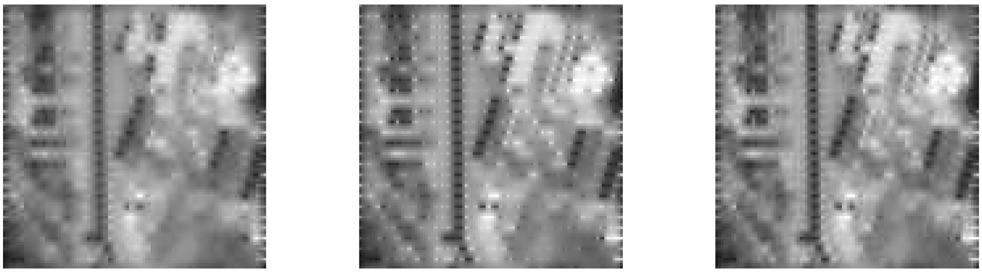


Figure 2.5: Uniform horizontal motion. Left to right: 2 (error 18.7%), 4 (error 16.4%), and 8 (error 16.4%) LR frames.

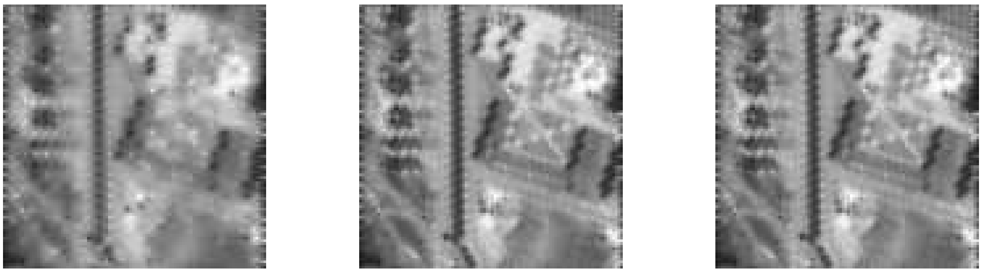


Figure 2.6: HR restoration with uniform diagonal motion. Left to right: 2 (error 18.2%), 4 (error 13.9%), and 8 (error 13.9%) LR frames.

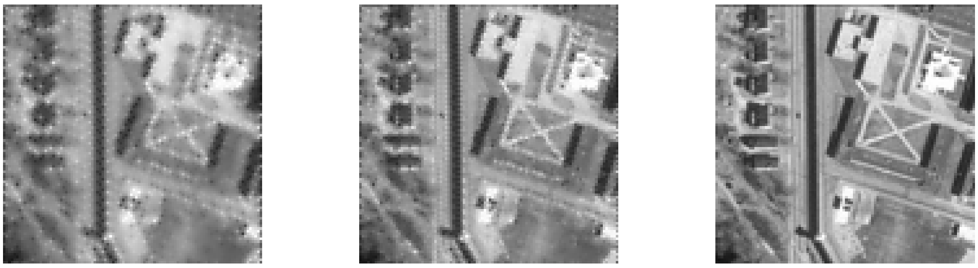


Figure 2.7: Zigzag motion. Left to right: 4 (error 12.7%), 8 (error 8.2%), and 16 (error 4×10^{-10}) LR frames.

Chapter 3

Motion Estimation

3.1 Introduction

Accurate image registration is essential in superresolution. As can be seen from the previous chapter, the matrix A depends on the relative positions of the frames. Motion estimation constitutes an extensive field of study by itself. Tom and Katsaggelos [10] state that "It is well-known that motion estimation is a very difficult problem due to its ill-posedness, the aperture problem, and the presence of covered and uncovered regions". In fact, the accuracy of registration is in most cases the limiting factor in HR reconstruction accuracy. The following are common problems that arise in estimating interframe displacements:

1. Local vs. global motion (motion field rather than a single motion vector). If the camera shifts and the scene is stationary, the relative displacement will be

global (the whole frame shifts). Typically, however, there are individual objects moving within a frame, from leaves of a tree swaying in the wind to people walking or cars moving. In such situations, it may be necessary to identify and determine the motion of each object individually.

2. Non-linear motion. Most motion that can be observed under realistic conditions is non-linear, but the problem is compounded by the fact observed 2-D image is only a projection of the 3-D world. Depending on the relative position of the camera and the object, the same object can appear drastically different. For example, a disc standing parallel to the image plane will appear as a circle. If it is rotated about a parallel axis, however, it will become an ellipse of shrinking width, until it finally looks like a line. Moreover, parts of an object may become invisible due to occlusion, and the rigid-body assumption may not hold—consider ripples on a person's shirt due to wind. If simple affine transformations, such as rotations on a plane, can theoretically be accounted for, there is no way to deal with changes in the object's shape itself, at least in non-stereoscopic models.

3. Changes in overall or local brightness of the scene. This commonly happens in IR cameras with automatic gain adjustment, but can also occur in optical cameras due to inherent reflectivity of surfaces (e.g. specular reflexes) or non-uniform illumination.

4. The "correspondence problem" and the "aperture problem", described in image processing literature, e.g. [32]. These arise when there are not features in an object being observed to uniquely determine motion. The simplest example would be an object of uniform color moving in front of the camera, so that its edges are not visible. Since the object is uniform, there is no way to tell how much it moved. Another example is an object with repetitive patterns, such as a brick wall. Since all bricks look the same, it might be difficult to estimate how far the camera actually moved with respect to the wall.

5. The need to estimate motion with subpixel accuracy. It is the subpixel motion that provides additional information in every frame, yet it has to be estimated from LR data. The greater the desired magnification factor, the finer the displacements that need to be differentiated (the two are inversely proportional).

6. The presence of noise. Noise is a problem because it changes the graylevel values randomly. To a motion-estimation algorithm, it might appear as though each pixel in a frame moves on its own, rather than uniformly as a part of a rigid object. In feature-based tracking, noise can obscure or hide important features. Sometimes, noise is particularly difficult to deal with because it affects both motion estimation and reconstruction.

Due to these difficulties, some authors do not consider the problem of registration at all, assuming the shifts are already know, as in controlled camera motion. However, this does not work for practical applications. In the next sections, we will discuss integer- vs. fractional-pixel motion and give a brief overview of the estimation techniques.

3.2 Integer-pixel Motion

From the HR restoration point of view, it is only the subpixel shifts that may contribute new information. Recall that in the forward model, we assume that LR data is generated by a series of shifts and subsampling. Fig. 3.1 shows a HR grid (1-D case is shown for simplicity), where the LR pixels remaining after downsampling by 3 are marked in black. If the data is shifted by 3 HR pixels (corresponding to 1 LR pixel) and downsampled, it will be no different from the first data set, except for an edge pixel. Therefore, shifts by an integer number of LR pixels produce redundant data. In an ideal case, it will simply produce a HR image that does not change with additional frames. Images taken under real conditions are at least slightly different from each, due to noise, even if the estimated displacement between them is an integer. The resulting HR image will then reduce the noise by averaging (for additive, shift-invariant noise). However, if the discrepancy is too great, it may produce objectionable artifacts.

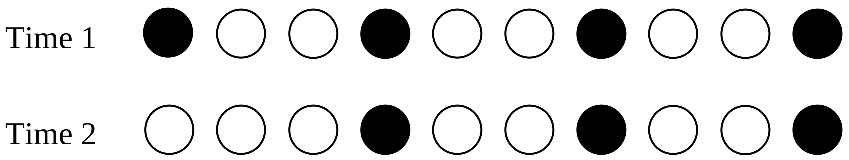


Figure 3.1: A shift by an exactly one LR pixel.

Despite the fact that we are interested in subpixel shifts, we cannot compute them in isolation. Image registration techniques designed for subpixel accuracy only work if the images are displaced by, indeed, a subpixel amount. Therefore, we must align the images as closely as possible, and this is why we need an integer-pixel estimator. We describe a few common ones next.

1) *Sum of squared differences (SSD).* Perhaps the most conceptually simple technique, it works by defining some distance measure between blocks in two frames (or, if motion is global, between the frames themselves). One of the blocks is fixed, while the for the other, the boundaries are shifted within some search region. After each shift, the distance measure between blocks is computed, and the displacement is determined by the location where this distance is minimized. Most commonly, the distance measure is the sum-of-square differences (SSD). For two images $f(x, y)$ and $g(x, y)$, it is defined as

$$SSD(d_1, d_2) = \sum_{i=-n_1}^{n_1} \sum_{j=-n_2}^{n_2} (f(x+i, y+j) - g(x+i-d_1, y+j-d_2))^2 \qquad (3.2.1)$$

where the summation extends over the region of size $(2n_1 + 1) \times (2n_2 + 1)$. Region matching is sensitive to noise, repetitive patterns, and is computationally expensive.

2) *Spatial cross-correlation.* The normalized correlation in the spatial domain is computed for two regions under consideration. Formally,

$$C(d_1, d_2) = \frac{\sum_{i=-n_1}^{n_1} \sum_{j=-n_2}^{n_2} (f(x+i, y+j) - \bar{f})(g(x+i+d_1, y+j+d_2) - \bar{g})}{\sqrt{\sum_{i=-n_1}^{n_1} \sum_{j=-n_2}^{n_2} (f(x+i, y+j) - \bar{f})^2 \sum_{i=-n_1}^{n_1} \sum_{j=-n_2}^{n_2} (g(x+i+d_1, y+j+d_2) - \bar{g}}} \tag{3.2.2}$$

where $\bar{f}$ and $\bar{g}$ are the averages of f and g [33].

Cross-correlation is conceptually similar to SSD. Both are block-matching techniques, only the distance measure is different. However, cross-correlation is not sensitive to overall changes in brightness because of the normalization.

3) *Phase correlation.* Let $g(x, y) = f(x - x_0, y - y_0)$, meaning the second image is obtained by shifting the first one by (x_0, y_0). Let $F(u, v)$ and $G(u, v)$ be the Fourier transforms of the images. According to the Fourier shift property,

$$G(u, v) = F(u, v) \exp[-j(ux_0 + vy_0)] \tag{3.2.3}$$

Then the cross-power spectrum is given by

$$C(u, v) = \frac{F(u, v)G^*(u, v)}{|F(u, v)G^*(u, v)|} = \exp[-j(ux_0 + vy_0)] \tag{3.2.4}$$

where the asterisks denote conjugation [24]. The easiest way to solve this for (u_{x_0}, v_{y_0}) is to take the inverse Fourier transform of $C(u, v)$, which should produce a Dirac delta function at (x_0, y_0).

Fig. 3.2 shows two images displaced by (9, 5) pixels with respect to each other and downsampled by 2 (both images are downsampled). Fig. 3.3 shows the plots

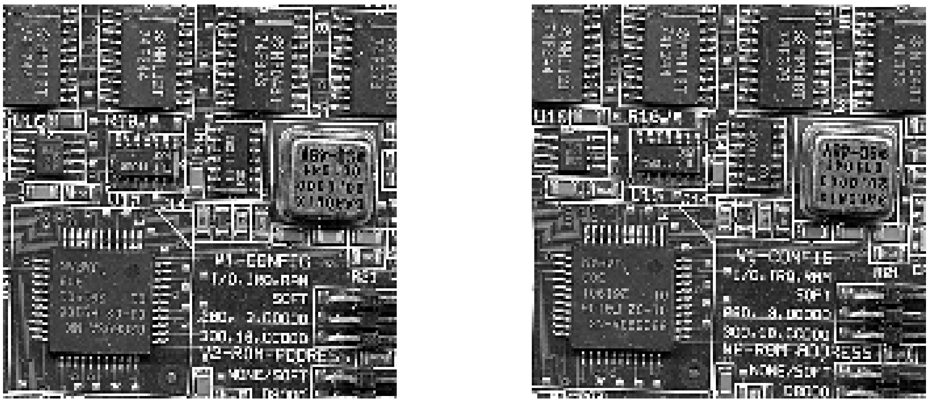


Figure 3.2: Two identical but shifted images.

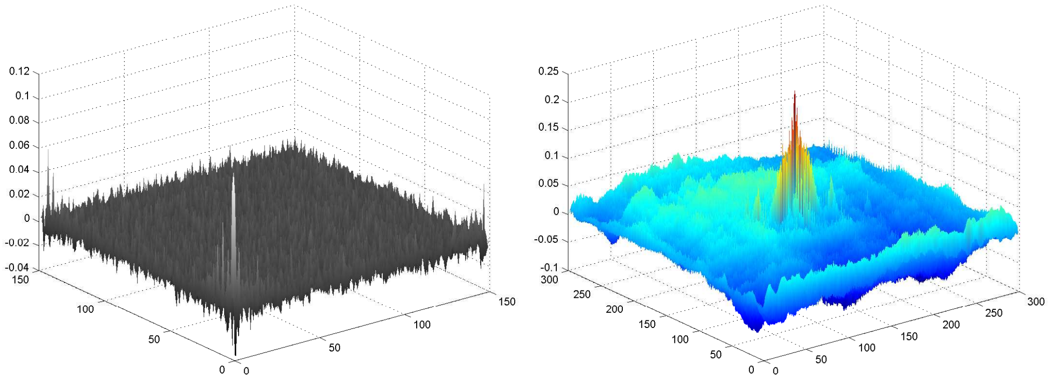


Figure 3.3: Plot of their IFT of phase correlation (left) and of spatial correlation (right).

of their phase correlation and normalized cross-correlation. The peak obtained with phase correlation is much sharper and better-defined. It is centered at (4, 2), which is the equivalent shift (to the nearest integer) in LR pixels after downsampling. Note also the presence of secondary peaks around the main peak. This is a consequence of downsampling and can be exploited to obtain fractional-pixel accuracy, as explained in the subpixel motion estimation section.

4) *Variational Methods.* This is a class of techniques based on the principles of

variational calculus. By Hamilton's principles, motion path is such that the integral of the Lagrange function L is minimized. For a point mass, this integral is

$$\int_{t_1}^{t_2} L(x, \dot{x}, t)dt \tag{3.2.5}$$

In the general case, the motion is a vector field $\mathbf{v}(\mathbf{x})$, a two-dimensional vector function of a two dimensional vector variable. The variation integral is

$$\int_{\text{window}} L(\mathbf{v}, \frac{\partial v_i}{\partial x_j}, \mathbf{d})d^2x \tag{3.2.6}$$

This problem can be solved by Euler-Lagrange equations. Details can be found in [32].

3.3 Subpixel Motion

In integer-pixel shifts, a pixel in location A in the first frame moves to location B in the second frame. Except near the image's edges, all pixels can be matched one-to-one between frames. In subpixel shifts, there is no one-to-one correspondence. Instead, the value of each pixel changes. For most pixels, this change is usually small and depends on the values of the neighboring pixels. Subpixel motion can be represented by a combination of integer-pixel shifts, blur (sometimes), and downsampling. Although in real images, the image formation process is different, this representation can still serve as a model of reasonable accuracy.

Subpixel displacement can be determined by methods described in the previous section, extended by interpolation. For example, spatial cross-correlation can be extended by upsampling both images and then computing their cross-correlation. Multiscale, or pyramid, methods,use resampling and interpolation at several levels. The results, however, are limited in accuracy and may be highly dependent on the interpolation used during upsampling (e.g. nearest-neighbor, bilinear, or bicubic). Here we briefly describe several additional methods for subpixel registration.

Phase correlation. While interpolation can also be used to extend phase correlation to subpixel shifts, there is a more efficient methods proposed by Foroosh and Zerubia [23]. The derivation can be found in their paper, and here we only present the final result. For an image pair subsampled by (r_x, r_y), the phase correlation can be approximated by

$$C(x,y) = \frac{\sin(\pi(r_x x - x_0))}{\pi(r_x x - x_0)} \frac{\sin(\pi(r_y y - y_0))}{\pi(r_y y - y_0)} \tag{3.3.1}$$

By applying this equation to the secondary peaks of phase correlation, the displacement (x_0, y_0) can be estimated. The authors note that the underlying assumption is that there is no aliasing during downsampling.

Gradient constraint. Let $f(x,y,t)$ be a time-varying image (i.e. an image sequence). By Taylor expansion, dropping the higher-order terms, we obtain

$$f(x + \triangle x, y + \triangle y, \triangle t) \approx f(x,y,0) + \nabla f(x,y,0)^T \triangle \tag{3.3.2}$$

where $\triangle = [\triangle x, \triangle y, \triangle t]^T$.

Therefore, for two consecutive frames, $f(x,y)$ and $g(x,y)$,

$$g(x,y) \approx f(x,y) + \frac{\partial f(x,y)}{\partial x}\triangle x + \frac{\partial f(x,y)}{\partial y}\triangle y \tag{3.3.3}$$

Therefore, to determine the shift $(\triangle x, \triangle y)$, we need to minimize the difference between the two sides of Eq. 3.3.3.

$$(\triangle x, \triangle y) = \arg \min_{\triangle x, \triangle y} \sum_x \sum_y \left(g(x,y) - f(x,y) - \frac{\partial f(x,y)}{\partial x}\triangle x - \frac{\partial f(x,y)}{\partial y}\triangle y \right)^2 \tag{3.3.4}$$

In matrix form, the solution to this is

$$\begin{bmatrix} \sum_x \sum_y \left(\frac{\partial f(x,y)}{\partial x}\right)^2 & \sum_x \sum_y \frac{\partial f(x,y)}{\partial x}\frac{\partial f(x,y)}{\partial y} \\ \sum_x \sum_y \frac{\partial f(x,y)}{\partial x}\frac{\partial f(x,y)}{\partial y} & \sum_x \sum_y \left(\frac{\partial f(x,y)}{\partial y}\right)^2 \end{bmatrix} \begin{bmatrix} \triangle x \\ \triangle y \end{bmatrix} = \begin{bmatrix} \sum_x \sum_y (g(x,y) - f(x,y))\frac{\partial f(x,y)}{\partial x} \\ \sum_x \sum_y (g(x,y) - f(x,y))\frac{\partial f(x,y)}{\partial y} \end{bmatrix} \tag{3.3.5}$$

Eq. 3.3.5 is called the gradient constraint equation because motion is constrained by the continuity of optical flow. It works well under the assumption that motion is smooth and less than one pixel.

3.4 Examples of Motion Estimation

The approach used in this project is to estimate the integer-pixel displacement using phase correlation, then align the images with each other using this estimate, and finally compute the subpixel shift by the gradient constraint equation. Fig. 3.4 shows two aerial photographs with a shift of (8, 13), downsampled by 3 in both directions. The output of the phase-correlation estimator was (3, 4), which is $(8, 13)/3$ rounded

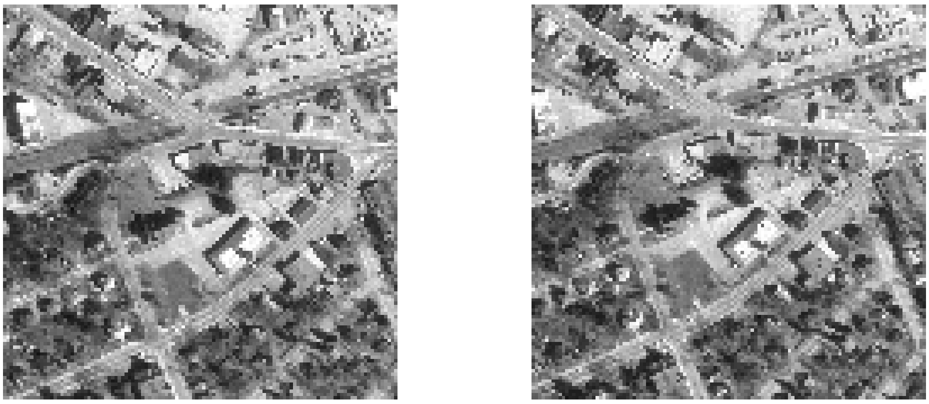


Figure 3.4: Image pair with a relative displacement of (8/3, 13/3) pixels.

to whole numbers. The second image was shifted back by this amount to roughly coincide with the first one (Fig. 3.5). Note that the images now appear to be aligned, but not identical, as can be seen from the difference image. Now the relative displacement between them is less than one pixel, and the gradient equation can be used. It yields $(-0.2968, 0.2975)$. Now, adding the integer and the fractional estimate, we obtain $(3, 4) + (-0.2968, 0.2975) = (2.7032, 4.2975)$. If this amount is multiplied by 3 and rounded, we obtain (8, 13). Thus we see that the estimate is correct.

3.5 Combinatorial Motion Estimation

Registration of LR images is a difficult task, and its accuracy may be affected by many factors, as described in the introduction. Moreover, in [25], it is stated that all motion estimators have inherent mathematic limitations, and in general, all of them are biased. A possible way to improve the accuracy is introduced in this section. The

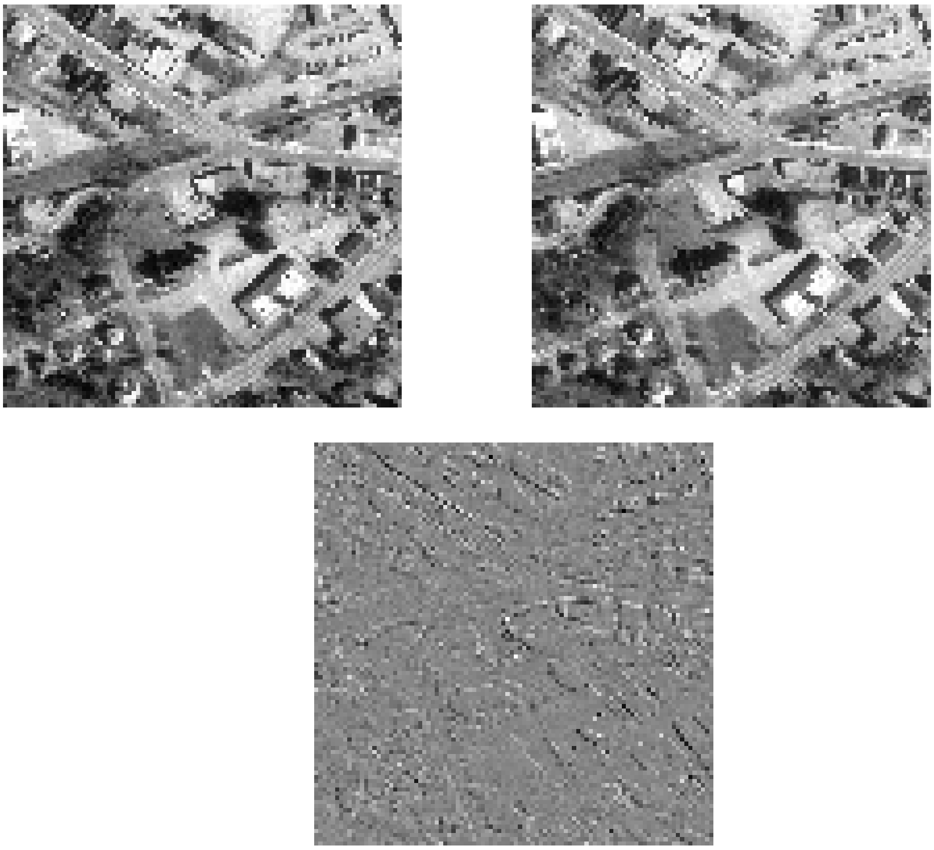


Figure 3.5: Images aligned to the nearest pixel (top) and their difference image (bottom).

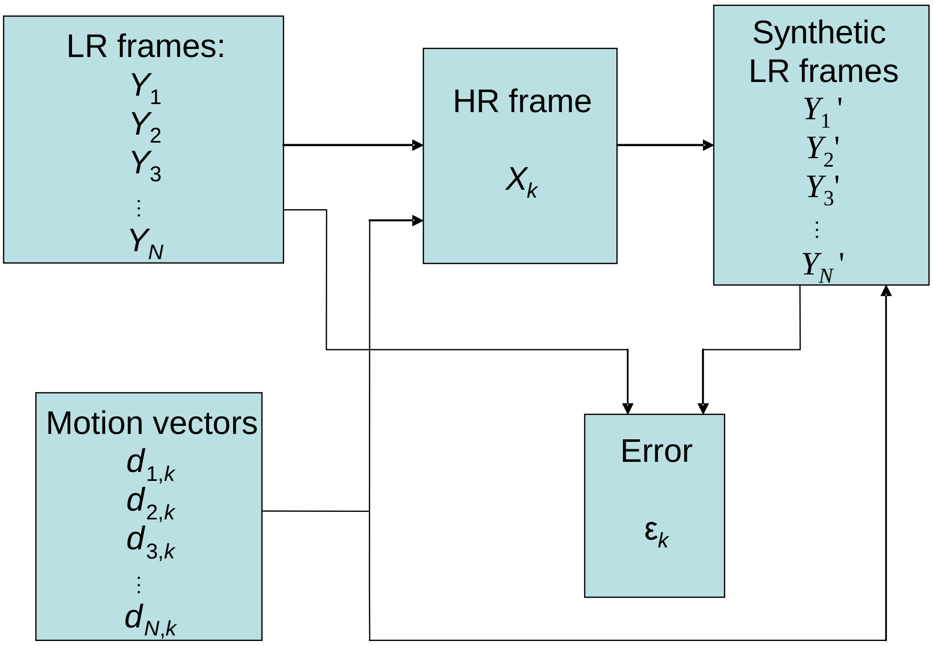


Figure 3.6: Block diagram of combinatorial motion estimation for case k.

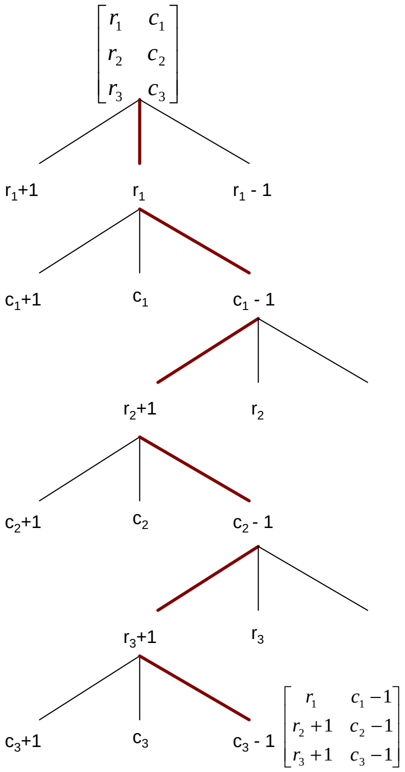
idea is to consider different possibilities for the motion vectors, and pick the best one. Since for real data, we do not know what a "good" HR image should like like, we define the best possibility as the one that best fits the LR data in the mean-square sense. So, having computed a HR image with a given set of motion vectors, we generate synthetic LR images from it and calculate the discrepancy between them and the real LR images. The same procedure is repeated, but with different motion vectors, and the motion estimate that yields the minimum discrepancy is chosen. Fig. 3.6 shows a schematic for evaluating the kth set of motion vectors.

Suppose we have N LR frames and $N-1$ corresponding motion vectors—one for each pair of adjacent frames. The vector for the shift between the first and the second frame is $d_{1,k}$, between the second and the third $d_{2,k}$, etc. The subscript k indicates that the motion vectors are not unique and we are considering one of the possibilities. Based on these vectors, we can generate both the HR image X_k and the LR images $\hat{Y}_{1,k}, \hat{Y}_{2,k}, \ldots, \hat{Y}_{N,k}$, where the circumflex is used to distinguish them from the real LR images $Y_{1,k}, Y_{2,k}, \ldots, Y_{N,k}$ (it is assumed that the upsampling/downsampling factor is constant for all k). The LR images can be converted into vector form, $y_{l,k} = \mathbf{vec}(Y_{l,k})$ and $\hat{y}_{l,k} = \mathbf{vec}(\hat{Y}_{l,k})$. The error (discrepancy) between the real and synthetic data is defined as

$$\varepsilon_k = \sum_{l=1}^{N-1} \frac{\|y_{l,k} - \hat{y}_{l,k}\|_2}{\|y_{l,k}\|_2} \tag{3.5.1}$$

Evaluating this equation for several motion estimates, we can choose the one that results in the smallest ε.

The list of different motion possibilities can come from different sources. For example, if there are several image registration algorithms, they may produce different outputs. Another way is to start with a coarse estimate and construct a search tree, where each branch represents a slightly different estimate. For example, consider a data set consisting of 4 LR frames and 3 associated motion vectors. Suppose that we know that in each frame, each vector may be off by at most one HR pixel horizontally and/or vertically. Then the search tree will consist of three levels (one per motion vector) and each children node will differ from its parent node by one. Part of this tree is shown in Fig. 3.7. After evaluating each branch, the correct sequence of motion



$\begin{bmatrix} r_1 & c_1 \\ r_2 & c_2 \\ r_3 & c_3 \end{bmatrix}$
r_1+1
r_1
$r_1 - 1$
c_1+1
c_1
$c_1 - 1$
r_2+1
r_2
c_2+1
c_2
$c_2 - 1$
r_3+1
r_3
c_3+1
c_3
$c_3 - 1$
$\begin{bmatrix} r_1 & c_1 - 1 \\ r_2 + 1 & c_2 - 1 \\ r_3 + 1 & c_3 - 1 \end{bmatrix}$

Figure 3.7: Part of a motion estimate search tree, with the correct path shown in bold.

vectors can be chosen, as shown in the figure.

With this approach, it is possible to refine the estimate provided by an image registration algorithm. For example, consider the aerial photograph used in the previous section. The following shifts were simulated in the image: $d_1 = (-2,-3), d_2 = (-4,-2), d_3 = (1,1)$, with downsampling by 3 in both dimensions after each shift. Thus 4 LR frames were obtained. Now, suppose those LR images were provided without a priori knowledge of the shifts. Suppose further that we were unable to determine the interframe displacements precisely and that the best estimate we have is given by $d_1 = (-1,-4), d_2 = (-3,-1), d_3 = (1,2)$. Note that this is slightly different from the true motion. Constructing a search tree as described and evaluating each branch, it is possible to recover the true motion, as verified by numerical simulation. Fig. 3.8 shows the original image, the image reconstructed from LR data using the correct motion vectors, and the reconstruction using the guess that was used as an input to the combinatorial algorithm. Although the reconstructed image in the center is of poorer quality than the original (which is logical, since there is not enough LR data for full restoration, as is usually the case), it is better than the one on the right. Quantitatively, the error is 10.7% in the "true motion" image, and 18.4% in the "estimated motion" image, compared to the original and using the Euclidean norm. This shows that for superresolution restoration, even small registration errors can have a big and adverse impact on quality.

Generally speaking, the combinatorial algorithm improves the initial guess, even if it cannot provide the "exact" result. The drawback is a huge increase in computational time. Instead of computing one HR image, which involves the inversion of a very large sparse matrix, a separate HR image has to be built for every final node in the tree. The number of possibilities grows exponentially with the number of initial motion vectors. In the case of the example described above, the total number of possibilities is $(3^2)^3 = 729$. The base 3 is due to the three options for each element of a vector—leave unchanged, add one, or subtract one. The power 2 is due to each

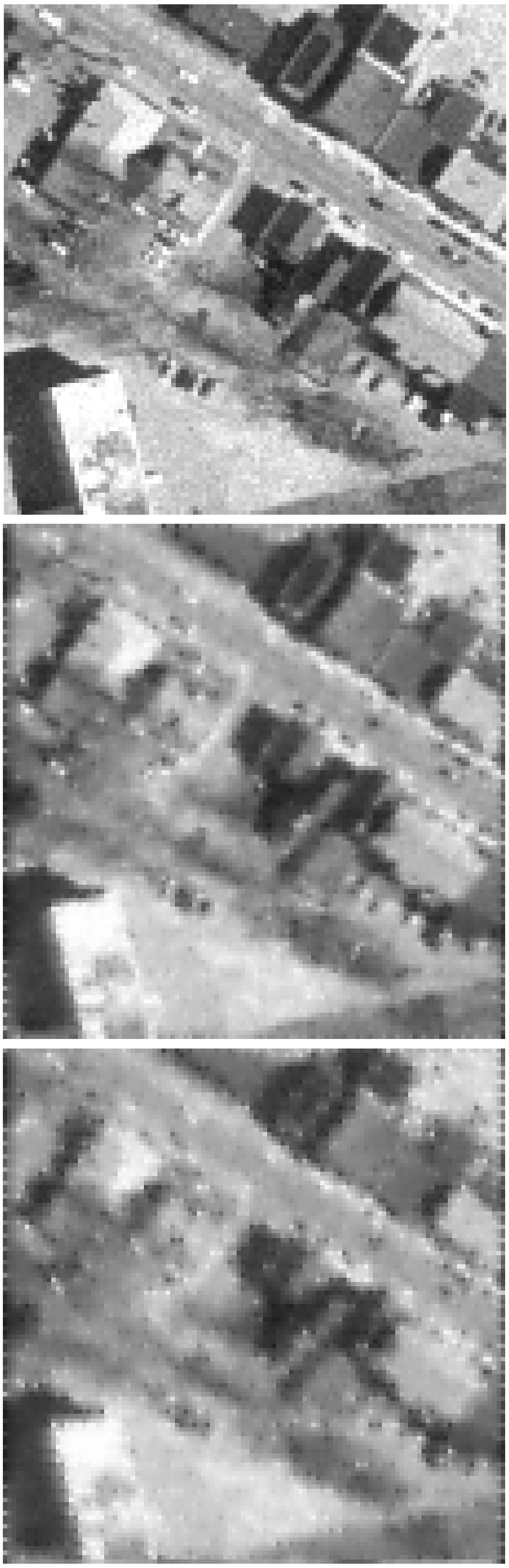


Figure 3.8: The original image (top), the restoration made with correct motion information (center), and the one made with an erroneous guess (bottom).

vector having two elements, corresponding to the horizontal and the vertical velocity components. The final power 3 represents the three vectors that we have to consider. It can thus be seen easily that the size of the tree grows with both the width and the depth of the search. For instance, if there were 4 motion vectors instead of 3, the number of possibilities would rise to $(3^2)^4 = 6561$. If, in addition, true velocity can differ from initial estimate by up to 2 pixels, it becomes $(5^2)^4 = 390625$.

It might be possible to reduce the search space by considering a "block of frames" at a time, i.e. for a 6-frame LR data set, apply the combinatorial algorithm to the first three and the last three frames independently. Alternatively, we may not consider the full path from the top to the bottom of the tree, but evaluate intermediate results at some point and continue tracing only the higher-ranked paths. More efficient methods, such as the branch-and bound algorithm, exist, but they are beyond the scope of this thesis.

3.6 Local Motion

Up to now, it has been assumed that the motion is global for the whole frame. Sometimes this is the case, for example when a camera is shaken randomly and the scene is static. In most cases, however, we are interested in tracking a moving object or objects. Even if there is a single object, it is usually moving against a relatively stationary background. One solution in this case is to extract the part of the frame that contains the object, and work with that part only. One problem with that approach is the boundary conditions. As described in Chapter 2, the model assumes that as the object shifts and part of it goes out of view, the new pixels at the opposite end are filled according to some predetermined pattern, e.g. all zeroes or the values of the previous pixels. In reality, of course, the pixels on the object's boundary do not change to zero when it shifts. This discrepancy does not cause serious distortions as long as the shifts are small relative to the object size. If all shifts are strictly subpixel, i.e. none exceeds one LR pixel from the reference frame, at most the edge pixels will be affected. However, as the shifts get larger, a progressively larger area around the edges of HR image is affected.

One solution is to create a "buffer zone" around the object and process this whole area. This is the region of interest (ROI) mentioned in the "Mathematical Description" chapter. In this case, when object's movement is modelled with shift operators, it is the surrounding area that gets replaced with zeroes, not the object itself. Since only the object moves, and the area around it is stationary, and we are treating all of ROI as moving globally, the result will be a distortion in the "buffer zone". However, we can disregard this since we are only interested in the object. In effect, the "buffer zone" serves as a placeholder for the object's pixels. It needs to be large enough to contain the object in all frames if the information about the object is to be preserved in its entirety. The only problem may be distinguishing between the "buffer zone" and the object (i.e. the object's boundaries) in the HR image, but this is usually apparent visually.

A second approach is to actually treat the background (buffer zone) as stationary, and only move the pixels in the object. This can then be naturally extended to the case of several objects, which is impossible with the first approach due to the assumption of global motion.

Naturally, for the second approach, the global shift operator cannot be used as before. The operator needs to act on the entire ROI, which consists of a moving object and stationary background. Instead, we define a "transition operator", which maps every pixel in frame A to a new location in frame B. The mapping does not have to defined by a single vector, and the mapping is one-to-one (this is ensured by choosing the ROI large enough so that no pixels "slide off the edge"). This mapping is given by

$$\begin{aligned}
i_T(i,j) &= \mathbf{vecind}(n_x n_y \times n_x n_y, i + d_x(i,j), j + d_y(i,j)) \\
j_T(i,j) &= \mathbf{vecind}(n_x n_y \times n_x n_y, i, j) \\
T(i_T(i,j), \forall j_T(i,j)) &= 0 \qquad (3.6.1) \\
T(i_T(i,j), j_T) &= 1 \\
T(j_T(i,j), \forall j_T(i,j)) &= 0
\end{aligned}$$

Here, T is the transition operator, (i,j) is the pixel's position in the original image, $(d_x(i,j), d_y(i,j))$ is the pixel's displacement, $(i + d_x, j + d_y)$ is the pixel's position in the new (displaced) frame, and the subscripts i_T, j_T are the rows and columns in T. The function **vecind** converts a matrix subscript to a vector index. It takes three arguments—the size of the matrix, the row subscript, and the column subscript. It returns the equivalent index of a vector that results from a lexicographically ordered matrix. In explicit form, it can be written as

$$\mathbf{vecind}(M \times N, i, j) = M(j-1) + i \tag{3.6.2}$$

The matrix T is of size $n_x n_y \times n_x n_y$, where $n_x \times n_y$ is the size of the image. T can be first constructed as an identity matrix (no displacement), and then some elements may be modified, corresponding to the shifted pixels. The notation $T(i_T, \forall j_T) = 0$ and $T(j_T(i,j), \forall j_T(i,j)) = 0$ in Eq. 3.6.1 means that the entire row of T is replaced with zeros. This is necessary to remove an existing pixel and replace it with a new one, in turn deleting the one in the original position. The product Ty of the transition matrix with a vectorized image gives a new image where each pixel may be in a new position.

The displacement $(d_x(i,j), d_y(i,j))$ is a function of (i,j) and therefore not restricted to be a single vector. Having started by considering a single object and its ROI, we can now see that the transition matrix can accommodate a more general model where each pixel can potentially have its own shift vector. Therefore, we can represent a ROI that contains multiple objects. Obviously, in this case the transition matrix must be recomputed for each frame. Recall that previously, we could build a single one-pixel shift operator and obtain multipixel shifts by raising it to a power equal to the length of the shift. Aside from that, the model of HR restoration is the same as for the global motion case. The subpixel shifts in the objects can be exploited to fill a single HR grid.

The local motion model has several limitations. First, the objects cannot occlude one another in any frame. If the assigned motion vectors cause objects to overlap, the transition matrix will preserve only one of the objects fully, and which one will depend on the order in which T is computed. Second, the objects are assumed to be of rectangular shape. Otherwise, instead of simply specifying the initial position of each object, a complete boundary descriptor would be necessary. Third, if the objects are far apart, or if they move far away from the original position, the ROI needs to be made large, even if the objects themselves are small. The size of the transition matrix, as well as other operators, grows as a square of the ROI size. So, even for a ROI of modest size, e.g. 100×100, the operators will be of size $10^4 \times 10^4$, and for, say, 5 LR images the HR restoration will involve the inversion of a $5 \cdot 10^4 \times 5 \cdot 10^4$ matrix. This is time-consuming even for sparse matrices. In cases like this, it may more convenient to process different objects separately, rather than combine them within a single ROI. Fourth, the motion vectors cannot be computed by simply providing a pair of frames as an input to a motion estimator. The objects have to be tracked individually for the entire LR sequence, and their initial locations need to be specified manually. Some image registration algorithms, for example, as used in certain types of video compression, partition the image into blocks and compute the shifts for each one. This is not applicable here, because this can split or merge individual objects and yield a very low-quality reconstruction. There exist methods for automatic segmentation, however, we do not consider them here. As a result of these limitations, superresolution with local motion is sometimes difficult in practice. Consider, for example, the silhouette of a walking person. The parts of the body that can be thought of separate blocks, such as head, body, arms, and legs, are neither rectangular nor occlusion-free. (In addition, human motion may be non-linear as mentioned before).

Fig. 3.9 shows an example of simulated local motion. It has three rectangular blocks with different initial positions, different sizes, and moving with different velocities. The two frames show the blocks' initial and final positions. It shows the limitations of the local motion model: the blocks have to be rectangular, they must not overlap, and the ROI needs to bound *all* objects in *all* frames (otherwise, it is impossible to construct a single transition matrix).

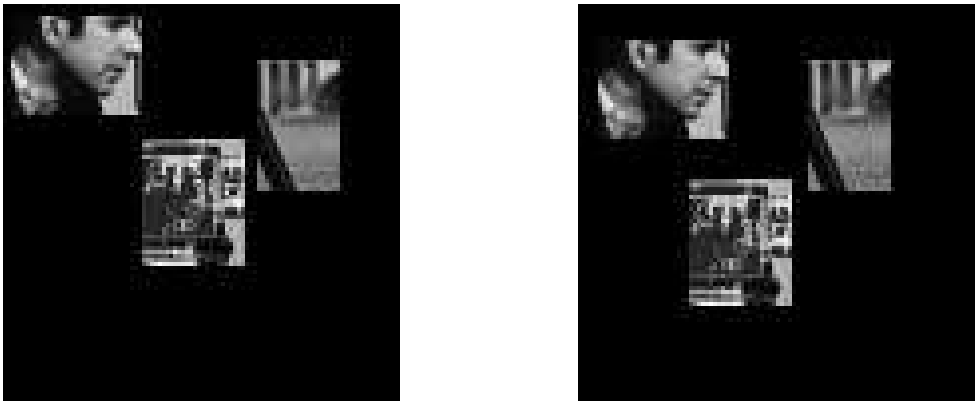


Figure 3.9: An example of local motion: three blocks of different sizes moving in different directions.

Chapter 4

Edge-Preserving Methods

4.1 Introduction

It has been mentioned before that in the proposed HR reconstruction scheme, there is a fundamental trade-off between smoothness of the superresolved image and the amount of noise or visually unappealing artifacts. This occurs because the solution penalizes discontinuities in the image. Discontinuities can be outlier pixels, restoration artifacts, or noise, but they can also be real edges that are present in the image. Different methods have been proposed to overcome this drawback. One of them is based on the L_1 norm, instead of the more common quadratic formulation. It is described in “Fast and robust multiframe super resolution” [19]. Other authors discuss the properties that the penalty (potential) functional must have so that it does not uniformly suppress edges and noise. For example, in [30], the authors propose that small gradients must be smoothed, while large gradients must be preserved. They also describe a class of functions that satisfy these conditions.

In the next section, we give a brief overview of an iterative image restoration algorithm proposed by Curtis Vogel. The reason that this particular algorithm was chosen is that, to our knowledge, it has not been applied to superresolution before. The mathematical theory behind the algorithm, including convex analysis, can be found in Vogel's text. We are primarily interested in how this method applies to superresolution and its performance compared to the method described previously in this thesis.

4.2 Total Variation Minimization by C. Vogel

The Total Variation Regularization by Curtis Vogel [31] is applicable in the general case of restoration of a degraded image where the degradation model is known.

Vogel defines total variation as

$$\mathrm{TV}(f) = \int_0^1 \left| \frac{df}{dx} \right| dx \tag{4.2.1}$$

with a generalization to two dimensions,

$$\mathrm{TV}(f) = \int_0^1 \int_0^1 |\nabla f| \, dxdy \tag{4.2.2}$$

This definition is unsuitable in numerical calculations due to the non-differentiability of the Euclidean norm at the origin. To overcome this, a small parameter β must be added to the norm, resulting in an approximation to TV(f):

$$J_\beta(f) = \int_0^1 \sqrt{\left(\frac{df}{dx}\right)^2 + \beta^2 dx} \tag{4.2.3}$$

or, for a 2-D case,

$$J_\beta(f) = \int_0^1 \int_0^1 \sqrt{\left(\frac{df}{dx}\right)^2 + \left(\frac{df}{dy}\right)^2 + \beta^2 dxdy} \tag{4.2.4}$$

Then, Vogel formulates the total variation problem in terms of minimizing the functional

$$T(f) = \frac{1}{2} \left\| Kf - d \right\|^2 + \alpha J(f) \tag{4.2.5}$$

where f is the reconstructed data, d is the original data, and K is the mapping matrix. This notation, adopted for consistency with Vogel's work, is analogous to the symbols A, x, and y used before. Note that this is the general formulation for reconstruction, with the first term representing discrepancy between the data and the estimate, and the second term a penalty that is a function of some properties of the estimate. In the case of Tikhonov regularization, this property is smoothness. In Vogel's method, J is a discrete approximation to Eqs. 4.2.3 and 4.2.4. The equation can be viewed as a cost function of the restored data; the best estimate is the one for which cost will be minimal.

For one-dimensional data, the discrete approximation to the penalty functional can be written as

$$J(f) = \frac{1}{2} \sum_{i=1}^{n} \psi\left((D_i f)^2\right) \Delta x \tag{4.2.6}$$

where D_i is the derivative operator, and ψ is a smooth approximation to twice the square root function with the property

$$\psi'(t) > 0 \text{ for } t > 0 \tag{4.2.7}$$

In the implementation of Vogel's algorithm, ψ was defined as

$$\psi(t) = 2\sqrt{t + \beta^2} \tag{4.2.8}$$

which is an approximation to Eq. 4.2.3.

The penalty functional is easily extended to two dimensions (assume f is now a matrix representing a function on a discrete grid):

$$J(f) = \frac{1}{2} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \psi \left(\left(D_{ij}^x f \right)^2 + \left(D_{ij}^y f \right)^2 \right) \tag{4.2.9}$$

where D_{ij}^x, D_{ij}^y are the discrete derivative operators in the x and y direction.

Based on total variation and applying methods of convex analysis, Vogel derives an iterative algorithm that can be applied to image restoration. It is called the primal-dual Newton's method for total variation-penalized least squares minimization in thwo space dimensions. Implementation of the algorithm in pseudo-code can be found in Algorithm 8.2.4 (Ch. 8) of the text. This algorithm was used to produce several superresolved images as shown in the next chapter.

Chapter 5

Experimental Results

In this chapter, we are going to present some superresolved images created by the algorithm described previously. For clarity, and to have a formal metric for comparisons, most of the image sequences in previous chapters were synthetic, that is, they were artificially generated from existing images. Here, we work with real data. The images are photographs, either infrared (IR) or optical, taken in the field. Since there is no "HR original", the reconstruction fidelity cannot be quantified. However, the difference between LR and HR can be seen visually.

Fig. 5.1 shows a side-by-side comparison of LR and HR images. The HR image was generated from portions of 9 LR photographs taken with a Kodak LS743 digital camera with a 4-megapixel resolution. For comparison purposes, the LR photograph was magnified by upsampling and bicubic interpolation, which produces the most

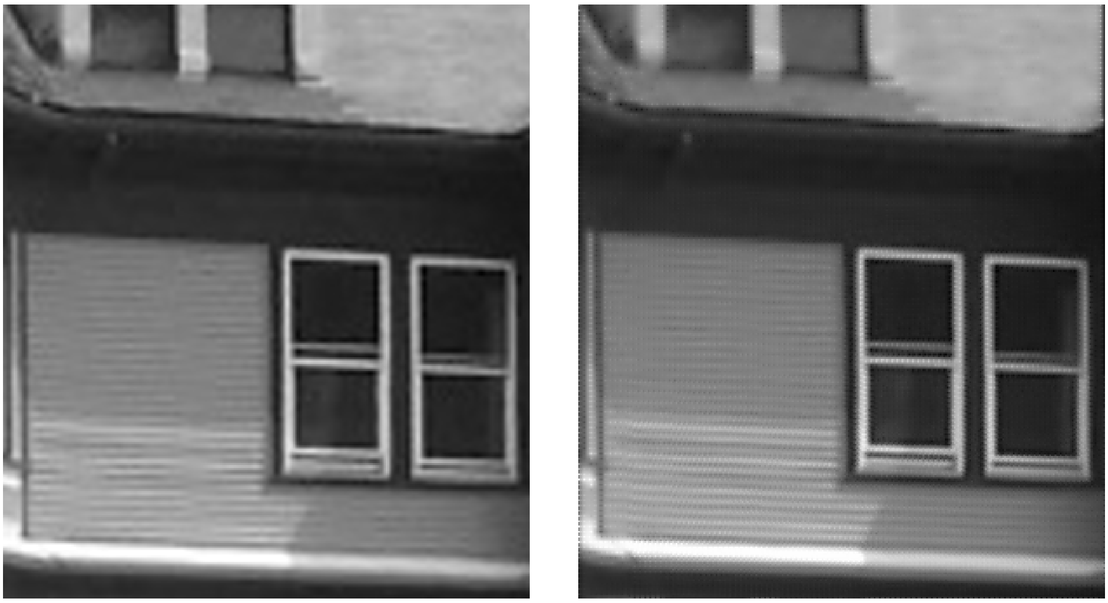


Figure 5.1: Magnified LR original (left) and HR restoration (right) of a wall of a house.

visually appealing result. The wood boards and windows are aliased in the LR photograph. Aliasing is significantly reduced in the HR image, which was computed with a 3x3 upsampling factor. There are more straight lines instead of broken lines. The reconstruction is not very good due to a slight shape distortion. This happens because the camera was handheld and the displacements between frames were generated manually moving it (in this case, the object, the wall, remains stationary). The displacements were not strictly translational, as the camera slightly twisted and turned sideways, and it was impossible to hold it perfectly still with the hands. Even a small rotation can adversely affect quality when images have to be aligned with subpixel accuracy. Still, the aliasing reduction demonstrates resolution enhancement under realistic conditions.

Fig. 5.2 shows a part of 8 consecutive frames of a video taken with an Indigo IR camera with a 160x120 array, wavelength of 7.5–13.5 micron, and a 30 mm lens (15x11 deg field of view (FOV)). (Note: this and all subsequent LR images were provided by Textron Systems). The full images, which show a person walking across the camera's FOV, were not necessary for processing since the goal is to obtain an HR image of the person, the surrounding objects and landscape are irrelevant. The processing was done only on the ROI. If, in one of the frames, the object slides past the edge of the ROI, that frame does not contribute to the restoration. In this case, as it can be seen, the person appears on the right side of the ROI in the first frame and the left side in the last one.

Fig. 5.3, on the left, shows the magnified version of the first frame for comparison purposes. On the right, the HR image (upsampling 2x2) obtained from the 8 LR frames is shown. Note the artifacts in the lower part of the image. This is due to the legs and arms moving differently from the head. The head was used for motion estimation, but human motion is not global and not purely translational. Therefore, the motion model for the lower body is wrong and results in strong distortions for that part. However, if we are interested in identifying the person, the face is of primary importance and the outside distortions can be disregarded. The head and shoulders are much sharper in the superresolved image, with the outline of the face clearly visible.

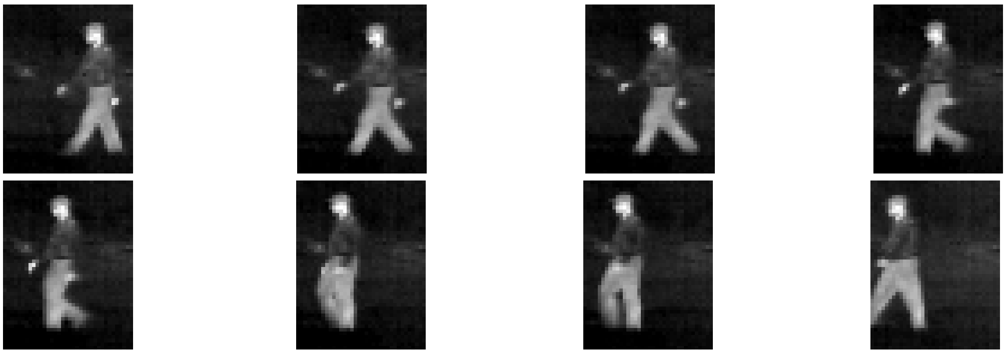


Figure 5.2: Eight frames in the LR sequence used for superresolution of a person.

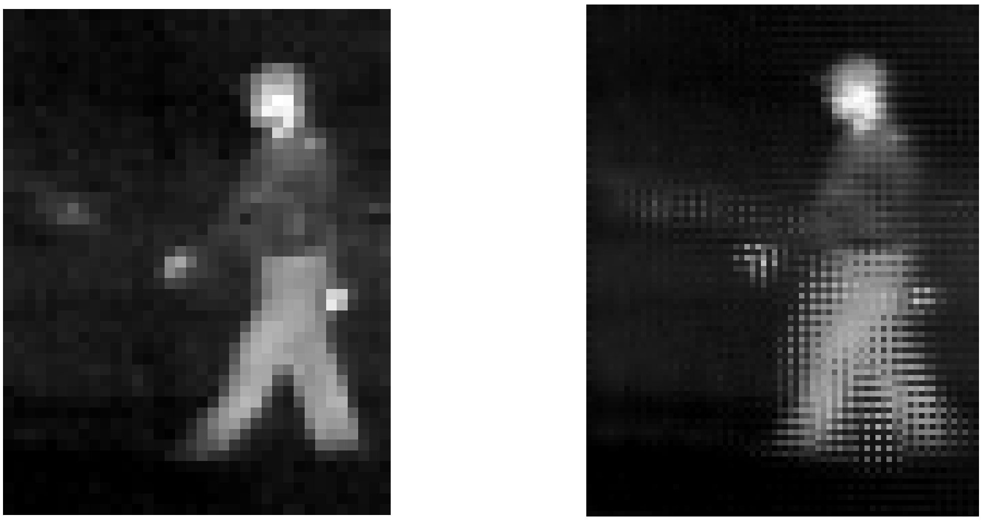


Figure 5.3: Magnified first LR frame (left) and the HR reconstruction of a walking person.

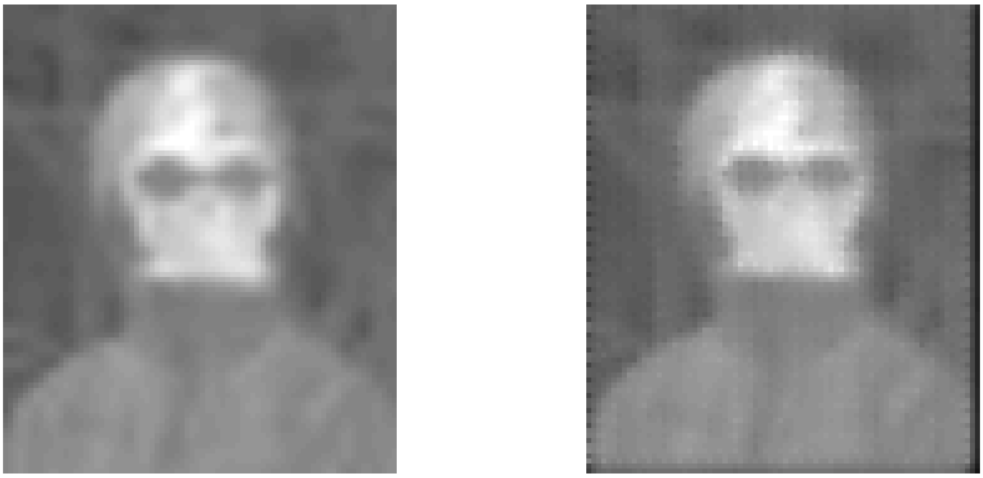


Figure 5.4: IR image of a person facing the camera: one of the original frames, upsampled (left) and the superresolved frame (right).

Fig. 5.4 shows another example of IR images where superresolution can help in identification. This time, the person is facing the camera and is relatively stationary (in the sense of not walking), but small changes in posture still produce subpixel displacements between frames. A total of 11 frames were processed and upsampled by 3x3 on a HR grid. The most obvious improvements are in the subject's head shape and the glasses.

On Fig. 5.5, an IR image of a parked car in front of a house is presented. As in the previous example, 11 frames and an upsampling factor of 3x3 were used. The aliasing present in the LR image is eliminated in HR reconstruction. Note the straight, sharp lines on the car and the house.

Superresolution processing can be applied to conventional as well as IR imagery. Fig. 5.6 shows a photograph taken on a highway with an optical Canon G3 camera

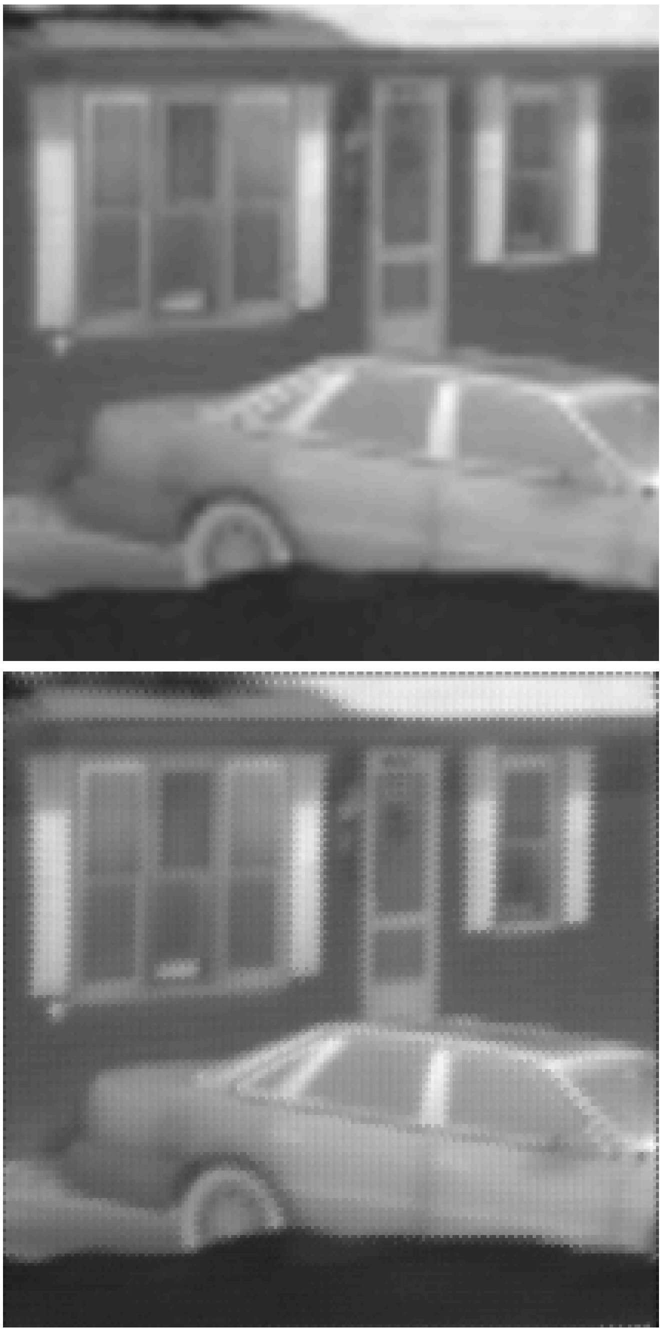


Figure 5.5: Another IR image example: upsampled LR with aliasing (top), higher-quality superresolved image (bottom).

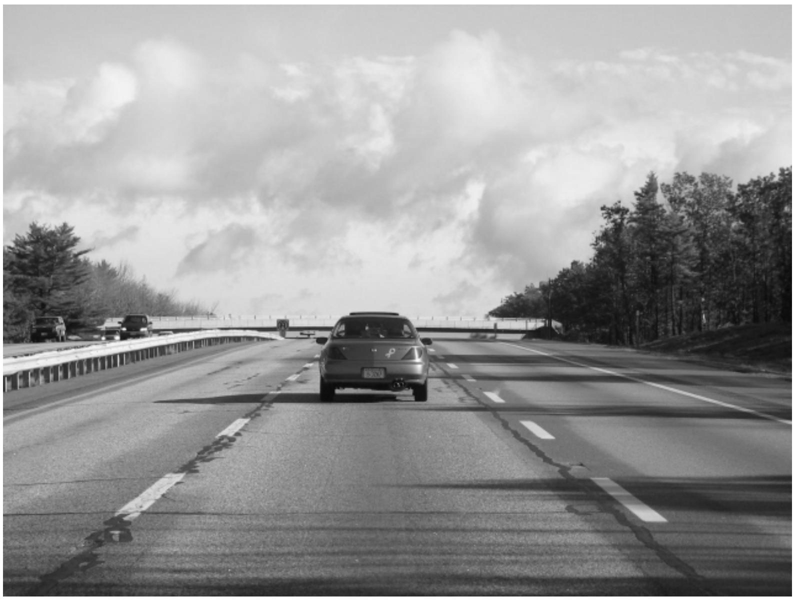


Figure 5.6: Full photograph of the scenery taken on a highway.

with 640x480 image size. Five such photographs were taken. Suppose we are interested in the license plate of the car ahead on the road. Fig. 5.7 shows 4 LR frames cropped to extract the ROI. At this resolution, the numbers on the plate are indistinguishable. Fig. 5.8 shows the superresolved frame with an upsampling factor of 3x3. The numbers are clearly readable. This example and the previous one demonstrate how superresolution can be useful in object identification and recognition. It can applied to areas such as surveillance, security, and tracking.

In the previous chapter, an image restoration method designed to preserve edges was discussed. Here, we show two examples of side-by-side comparison of the original (Tikhonov regularization) method and the primal-dual Newton's method based on total variation minimization. Fig. 5.9 shows the license plate image presented earlier.

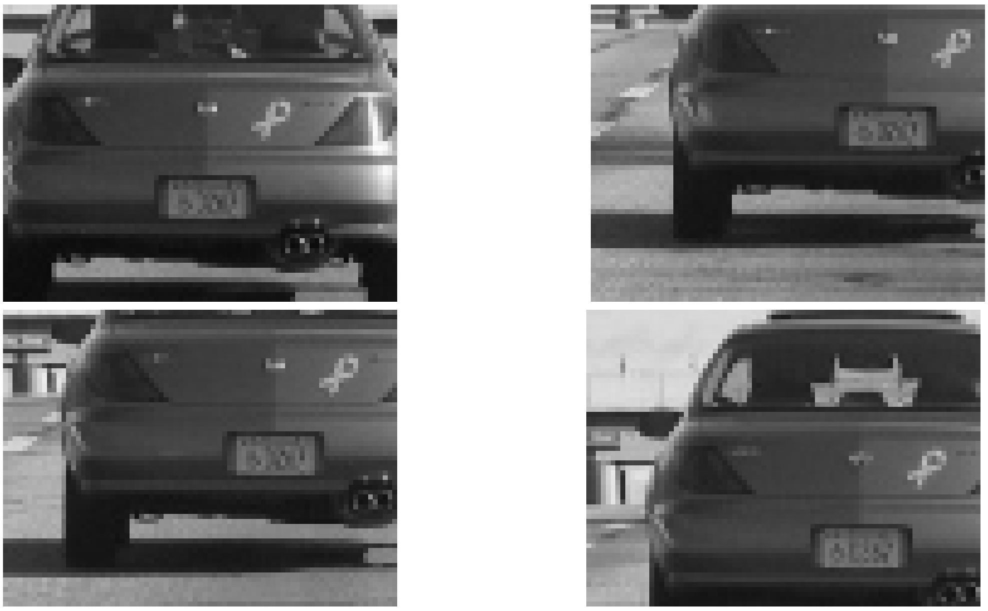


Figure 5.7: Four cropped frames used in reconstruction.

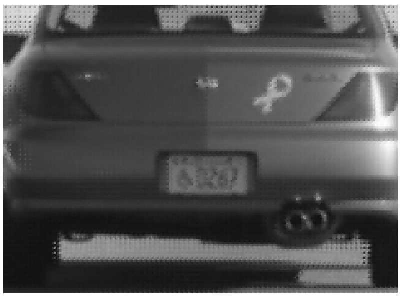


Figure 5.8: The superresolved image of the license plate.

The left side is the Tikhonov HR restoration and the right side is the HR image obtained by applying Vogel's edge-preserving algorithm. It can be seen that Vogel's technique offers no apparent improvements in this case. In fact, the letters on the license plate appear to be less legible. Fig. 5.10 shows another example where the opposite is true. The outline of the person's head is slightly sharper using Vogel's algorithm. This can be explained if we remember that edge-preserving methods tend

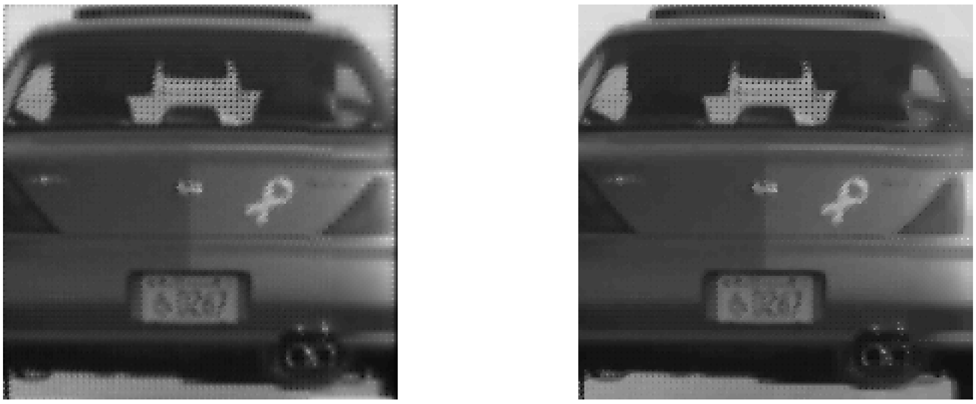


Figure 5.9: HR image of the license plate using Tikhonov regularization (left) and Vogel's total-variation-penalized primal-dual Newton's method (right).

to produce piecewise-continuous images; that is, images with pixel blocks where the brightness is constant within each block. This produces images with sharp edges, but it is poorly suited for images with a lot of grayscale variation. The license plate image has many shades of gray, while the IR image of a person is essentially black-and-white. Fig. 5.11 shows the cost function T of the Vogel algorithm. The cost is plotted as a function of the iteration index, for a total of 10 iterations. The cost is monotonically decreasing, as expected, and levels off after a few iterations.

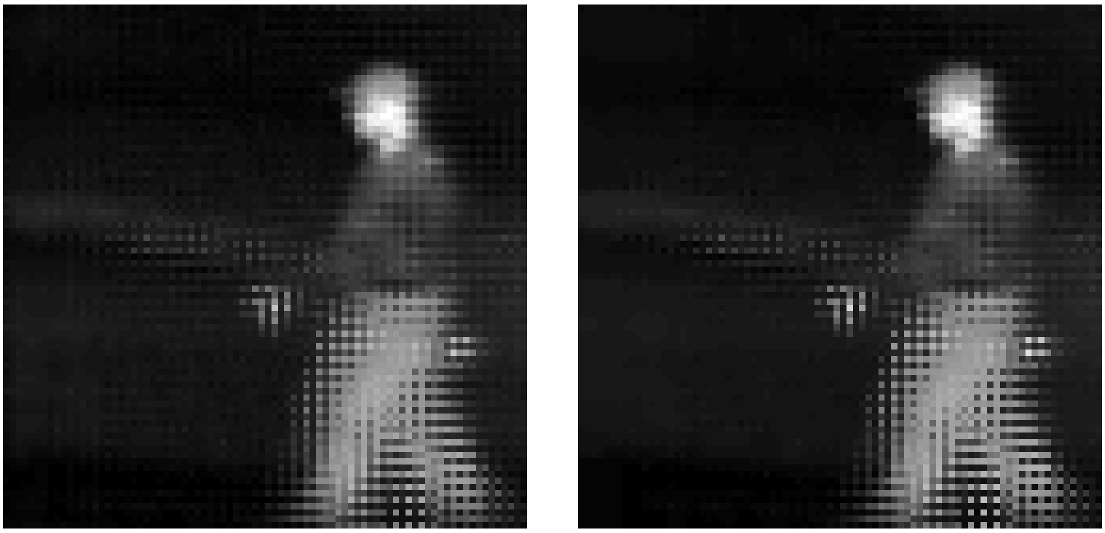


Figure 5.10: HR image of the person using Tikhonov regularization (left) and Vogel's total-variation-penalized primal-dual Newton's method (right).

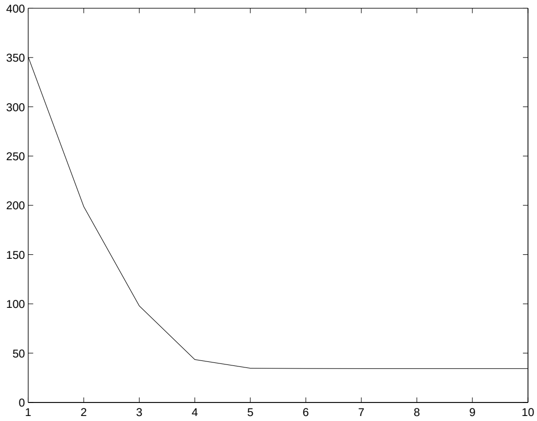


Figure 5.11: Cost function in Vogel's algorithm (horizontal axis represents the iteration number).

Chapter 6

Conclusions and Future Work

We have developed a matrix-based framework for obtaining high-resolution images from a low-resolution video clip or a sequence of displaced frames. The model describes the image degradation process in terms of matrix operations. The basic matrix blocks are translational shift, downsampling, and blurring operators, although other linear operations, such as rotation, can also be represented in terms of matrices. It is shown that the degradation process can be performed as a single matrix-vector multiplication. Then, the inverse process that reverses the image's degradation is described mathematically with the aid of Tikhonov regularization, which accomplishes an acceptable compromise between fidelity and sharpness of the image and the presence of undesirable artifacts. The feasibility of the model is shown theoretically, experimentally via computer simulation, and experimentally with real data.

The problem of registration, which is crucial for aligning the LR images to fuse them into one, is also addressed in this thesis. The method proposed here uses the well-known techniques of phase correlation and gradient constraint to achieve subpixel accuracy while being applicable to multipixel shifts. We presented a method for improving the accuracy of registration by combinatorial search, where the estimates are adjusted to match the given data.

The method described in the thesis can be applied to objects moving within a frame as well as whole frames. In the case of moving objects, the way to define a region of interest within the frame was shown. This allows to obtain a high-quality image of the object while eliminating the need to process irrelevant areas. The ROI for processing must be larger than the object, while the ROI for motion estimation is (typically) smaller, because it must only contain the pixels that are moving and not the stationary ones that surround the object.

We briefly discussed the primal-dual Newton's method developed by Curtis Vogel. This methods has the important advantage of preserving edges better. While this method was applied to deblurring problems by Vogel, we have shown experimentally that in certain cases, it produces good results for superresolution.

All methods and algorithms described in the thesis have been tested with both synthetic and real data. Superresolution can be applied to a great variety of problems, practically anywhere where imaging or computer vision is involved. However, an important application for ROI processing, in particular, is for object tracking and identification. This is a very common task for automated security and surveillance systems. For example, the superresolved IR images of a walking person shown in the Experimental Results chapter allow better recognition based on facial features. The license plate example shows how the originally unreadable numbers can be identified.

There are several areas in which future research may be performed. One is a more elaborate motion and degradation model. While non-linear effects, such as arbitrary rotation, shear, and deformation are difficult to include, rotation parallel to the image plane and motion blur (as opposed to sensor blur) can be added by forming additional matrix operators. A more precise blur model can also be obtained if the camera's PSF is known. A motion model incorporating multiple moving objects was developed in this thesis. However, it was not tested on real data due to constraints of rectangular boundaries for the objects and non-overlap. A more sophisticated model could allow non-rectangular objects, for example, by specifying a geometrical descriptor or vertices of a polygon. Alternatively, it could use automated segmentation and tracking, which would be another line of work. The local motion model could incorporate overlap during motion, or objects that are a part of another object. For complex motion patterns, such as a person walking or running, all of these factors need to be considered: local non-linear motion, complex boundary, and occlusion (e.g. an arm covering the body). Estimating and parameterizing such motion is a challenging problem by itself, even without constructing matrix operators for it.

Another area where a lot of work can be done is automating ROI selection and object tracking. Under the current scheme, the ROI coordinates for both HR restoration and motion estimation have to be specified manually with reference to the frame. There could be an algorithm that determines the boundaries of the object and the ROI based on what parts of the image appear to move consistently. If there are many non-stationary pixels in the frame, the algorithm might even select a ROI by knowing roughly what the object or objects look like (for example, if one of the objects resembles a human figure). This connects with the problem of automatic tracking and identification, where a system would identify possible targets, track them for several frames, providing an estimate of the position in each frame, then construct an HR image of the object and identify it or take some action depending on the object.

Finally, the question of edge preservation and reducing artifacts needs to be considered. Typically, sharper edges mean more details can distinguished in the object. The only way to sharpen edges in the Tikhonov method is to decrease the regularization parameter. However, this inevitably amplifies noise. This is particularly problematic when blur is included in the model, since deblurring is a high-pass filtering operation. When artifacts become objectionable, we are forced to increase the regularization parameter. Thus, we are sharpening the image by deblurring while smoothening it by the regularizer. There may be better ways to achieve a compromise between distortions and sharpness. While Vogel's edge-preserving algorithm was explored in this thesis, the results are worse compared to Tikhonov regularization for some images and only marginally better for others. Other edge-preserving methods exist, such as the one proposed by Farsiu, et. al. [19]. Perhaps a different method could be selected depending on the type of image; for example, Vogel's method performs well for piecewise-continuous IR images.

Bibliography

[1] Sean Borman, Robert Stevenson, “Spatial resolution enhancement of low-resolution image sequences: a comprehensive review with directions for future research”. 1998.

[2] R. Y. Tsai and T. S. Huang, “Multiframe image registration and restoration”. *Advances of Computer Vision and Image Processing*, vol. I, 1984.

[3] A. M. Tekalp, M. K. Ozkan, and M. I. Sezan, “High-resolution image reconstruction from lower-resolution image sequences and space-varying image restoration”. *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. III, 1992.

[4] S. P. Kim, N. K. Bose, H. M. Valenzuela, “Recursive reconstruction of high resolution image from noisy undersampled multiframes”. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 38, no. 6, 1990.

[5] S. P. Kim, N. K. Bose, H. M. Valenzuela, “Recursive total least squares algorithm for image reconstruction from noisy, undersampled multiframes”. *Multidimensional Systems and Signal Processing*, vol. 4, no. 3, July 1993.

[6] C. L. Luengo Hendriks, L. J. van Vliet, “Improving resolution to reduce aliasing in an undersampled image sequence”. *SPIE*, vol. 3965, Jan. 2000.

[7] Marc Rumo, Patrick Vandewalle, “Superresolution in images using optical flow and irregular sampling”.

[8] Richard R. Schultz, Robert L. Stevenson, “Extraction of high-resolution frames from video sequences”. *IEEE Trans. on Image Processing*, vol. 5, no. 6, June 1996.

[9] Brian C. Tom, Aggelos T. Katsaggelos, “Reconstruction of a high resolution image from multiple degraded mis-registered low resolution images”. *SPIE*, vol. 2308,

[10] Brian C. Tom, Aggelos T. Katsaggelos, “Resolution Enhancement of Video Sequences Using Motion Compensation”.

[11] “High-resolution image reconstruction from digital video by exploitation of non-global motion”. *SPIE*, vol. 38, no. 5, May 1999.

[12] Michael Elad, Arie Feuer, “Super-resolution reconstruction of image sequences”. *IEEE Trans. On Pattern Analysis and Machine Intelligence (PAMI)*, vol. 21, no. 9, September 1999.

[13] Michal Irani, Shmuel Peleg, “Improving resolution by image registration”. *CVGIP: Graphical Models and Image Processing*, vol. 53, no. 3, May 1991.

[14] Michal Irani, Shmuel Peleg, “Motion analysis for imae enhancement: resolution, occlusion, and transparency”. *Journal of Visual Communications and Image Representation*, vol. 4, Dec. 1993.

[15] N. R. Shah and A. Zakhor, “Multiframe spatial resolution enhancement of color video”. *Proc. of the IEEE Intl. Conference on Image Processing*, vol. I, Sep. 1996.

[16] R. C. Hardie, T. R. Tuinstra, J. Bognar, K. J. Barnard, and E. Armstrong, “High resolution image reconstruction from digital video with global and non-global scene motion”. *Proc. of the IEEE International Conference on Image Processing*, vol. I, Oct. 1997.

[17] R. C. Hardie, K. J. Barnard, and E. Armstrong, “Joint MAP registration and high-resolution image estimation using a sequence of undersampled images”. *IEEE Trans. on Image Processing*, vol. 6, no. 12, Dec. 1997.

[18] H. Stark and P. Oskoui, “High-resolution image recovery from plane-image arrys, using convex projections”. *Journal of the Optical Society of America A*, vol. 6, no. 11, 1989.

[19] Sina Farsiu, M. Dirk Robinson, Michael Elad, and Peyman Milanfar, “Fast and robust multiframe super resolution”. *IEEE Trans. on Image Processing*, vol. 13, no. 10, Oct. 2004.

[20] A. J. Patti, M. I. Sezan, and A. M. Tekalp, “High resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur”. *Proc. of the IEEE International Conference on Image Processing*, vol. I, 1994.

[21] Eric Kaltenbacher, Russell C. Hardie, “High resolution image reconstruction using multiple, low resolution, aliased frames”. *SPIE*, vol. 2751, 1996.

[22] Mohammad S. Alam, John G. Bognar, Russell C. Hardie, Brian J. Yasuda, “High resolution image reconstruction using multiple, randomly shifted, low resolution, aliased frames”. *SPIE*, vol. 3063, 1997.

[23] Hassan Foroosh, Josaine B. Zerubia, “Extension of phase correlation to subpixel registration”. *IEEE Trans. on Image Processing*, vol. 11, no. 3, March 2002.

[24] Hassan Foroosh, Scott Hoge, “Motion information in the phase domain”. in *Video registration* (editors M. Shah and R. Kumar), vol. 5 of Kluwer Intl Series in Video Computing, ch. 3, Kluwer Academic Publishers, May 2003.

[25] Dirk Robinson, Peyman Milanfar, “Fundamental performance limits in image registration”. *IEEE Trans. on Image Processing*, vol. 13, no. 9, Sep. 2004.

[26] Nhat Nguyen, Peyman Milanfar, and Gene Golub, “A computationally efficient superresolution image reconstruction algorithm”.

[27] Michael K. Ng and Nirman K. Bose, “Mathematical analysis of super-resolution methodology”. *IEEE Signal Processing Magazine*, May 2003.

[28] Zhouchen Lin, Heung-Yeung Shum, “Fundamental limits of superresolution algorithms under local translation”. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, Jan. 2004.

[29] Eric Miller, *Inverse Problems* (course notes).

[30] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud, “Deterministic Edge-Preserving Regularization in Computed Imaging”. *IEEE Trans. on Image Processing*, vol. 6, no. 2, Feb. 1997.

[31] Curtis R. Vogel, *Computational Methods for Inverse Problems.* SIAM, 2002.

[32] Bernard Jähne, *Digital image processing: concepts, algorithms, and scientific applications.* Springer-Verlag, 3rd edition, 1995.

[33] Emanuele Trucco, Alessandro Verri, *Introductory Techniques for 3-D Computer Vision*. Prentice-Hall, 1998.