UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ESTIMATION THEORETIC ANALYSIS OF MOTION IN IMAGE SEQUENCES

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

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

M. Dirk Robinson

December 2004

The Dissertation of M. Dirk Robinson
is approved:

Professor Peyman Milanfar, Chair

Professor Benjamin Friedlander

Professor Michael Elad

Robert C. Miller
Vice Chancellor for Research and
Dean of Graduate Studies

Contents

List of Figures

List of Tables

Abstract

Estimation Theoretic Analysis of Motion in Image Sequences

by

M. Dirk Robinson

Estimating the motion (or dynamics) manifested in a set of images or an image sequence is a fundamental problem in both image and video processing and computer vision. From a computer vision perspective, much of what is interpretable in any real-world scene is reflected in the apparent motion. For instance, estimating the apparent motion in a video sequence provides the necessary information for many applications including autonomous navigation, industrial process control, 3-D shape reconstruction, object recognition, robotic motion control, object tracking, and automatic image sequence analysis. In image and video processing, the estimation of motion plays a vital role in video compression as well as multi-frame image enhancement. Disparate as they may seem, these many applications share one common thread: in all such applications, the demand is high for accurate estimates of motion requiring minimal computational cost.

In this thesis, we offer an estimation theoretic perspective on the problem of estimating motion from an image sequence. In particular, we focus on the various performance tradeoffs in both accuracy and computational efficiency associated with motion estimation. It is our goal that this work provide a common framework with which to evaluate and understand motion estimation performance.

To this end, this thesis offers contributions in three main areas. The first contribution is the proposal of a mechanism to greatly reduce the computational complexity in estimating complex motion vector fields from image sequences. In particular, we develop novel algorithms for estimating motion vector fields using tomographic projections. For example, we show that by incorporating tomographic projections into a multiscale gradient-based algorithms, we may achieve dramatic computational speedups while sacrificing little in the way of estimator accuracy. The second contribution is a thorough analysis of the widely popular class of gradient-based motion estimation algorithms. We derive and analyze the bias for this class of estimators and propose novel methods for optimizing gradient-based estimator performance. The third contribution is the analysis of the fundamental bounds limiting the accuracy of motion estimation. Specifically, we study the Cramér-Rao bounds associated with the problem of estimating translational motion in both aliased and non-aliased images. Finally, we show the intimate relationship between the performance bounds for motion estimation of aliased images and the problem of multi-frame image reconstruction.

Acknowledgements

The work contained in this dissertation represents the accumulation of many years of work made possible only by the collective support of family, friends, colleagues and mentors.

I would like to express my deep gratitude to my advisor, Professor Peyman Milanfar, for his patient guidance, commitment to my growth and development, and most importantly, for his example of professionalism in the field of engineering. I would like to thank Michael Elad whose keen insight and boundless energy proved to be a tremendous resource. I would also like to thank Professor Benjamin Friedlander for his perspective and observations which have helped improve the content of this work.

I would like to acknowledge the friends and family who have provided immeasurable support throughout this journey. Thanks to Dr. Jin, Sina, Morteza, Amyn, LiRui, and Saar for keeping The Lab an educational and interesting place to work. Thank you to my parents whose love and encouragement helped keep me on track for so many years.

Most importantly, I would like to thank Emily for your love and support which is what made this possible.

Dedicated to the memory of my grandfather H.S. Stout.

Chapter 1

Introduction

Estimating the motion (or dynamics) manifested in a set of images or an image sequence is a fundamental problem in both image and video processing and computer vision. For instance, a goal of computer vision is that of enabling a computer system to *interpret* the world using visual information sensed using a video imaging systems. Much of what is interpretable in any scene is reflected in the apparent motion. For instance, estimating the apparent motion in a video sequence provides the necessary information for many applications including autonomous navigation, industrial process control, 3-D shape reconstruction, object recognition, robotic motion control, object tracking, and automatic image sequence analysis [2–9]. In the field of video coding, the predictive power of accurate motion estimation is used to compress video sequences [10–12]. In image sequence processing, accurate motion estimates are used to improve overall image resolution. Disparate as they may seem, these many applications share one common thread: in all such applications, the demand is high for accurate estimates of motion requiring minimal computational cost. Therefore, numerous algorithms have been developed over the years to address the problems associated with motion estimation.

Because these high-level imaging applications are increasingly more pervasive in today's society, understanding the issues relating to performance is essential to build dependable and predictable systems. Generally, image processing applications are complex due to the large quantities of information present in the form of two and even three dimensional data signals. As such, the design and construction of motion estimation algorithms naturally offer substantial flexibility in trading off the computational complexity of a motion estimation algorithm with overall estimator accuracy. In one particular class of applications, the ideal tradeoff is one which sacrifices minimal accuracy to realize substantial gains in computational complexity. For example, in real-time motion compensated video encoders, the computational efficiency of motion estimation is critical. In fact, most real-time video coders require special hardware to achieve the motion estimation efficiency necessary to support real-time encoding [13]. For other applications, such as super-resolution, motion estimation accuracy is preferred without regard to the computational expense. Whatever the application, it is important not only to utilize such algorithmic flexibility but to understand the implicit associated tradeoffs.

Fundamental limits to estimator accuracy play a vital role in the analysis and development of algorithms. The ideal performance limits offer the measuring stick with which to objectively evaluate a host of algorithms. Furthermore, such limits suggest not only the need for further improvement, but also suggest when particular problems are effectively solved. In addition, the analysis needed to derive performance limits often generates significant insight into the performance bottlenecks associated with a given task. Finally, performance bounds on particular estimation problems provide understanding critical to the design of high level applications which rely on such lower-level estimation.

1.1 Introduction to Motion Estimation

In this section, we describe the models used to define the class of motion estimation problems we analyze in this thesis. We suppose that the imaging system provides measurements of the image intensity function $f(x_1, x_2, t)$ which represents the light emanating from the observed scene impinging on the 2-D focal plane of the imaging sensor. In this formulation, the terms x_1 and x_2 represent the spatial coordinates in this image sensor plane and t the time vari-

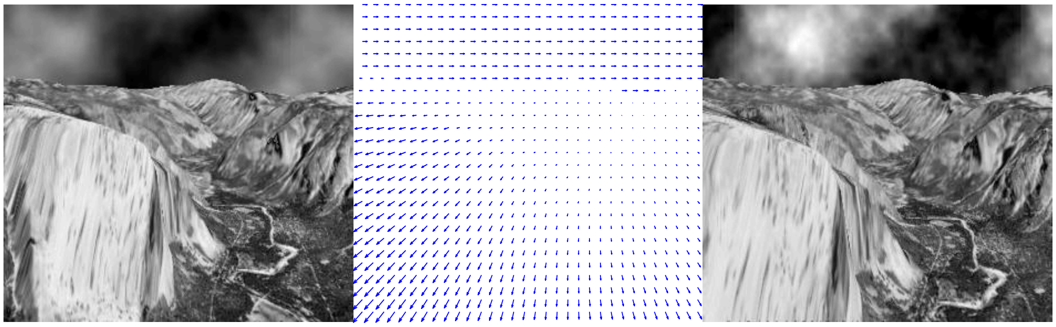

Figure 1.1: Example of a velocity vector field $\mathbf{v}(x_1, x_2)$ for the Yosemite Sequence

able. The image intensity functions we consider are modelled as temporally evolving according to

$$f(x_1, x_2, t) \quad = \quad f(x - v_1(x_1, x_2, t), x_2 - v_2(x_1, x_2, t), 0) \tag{1.1}$$

which is also known as the Intensity Conservation Assumption [2]. Such an assumption states that the intensity function for a given region remains the same even if the location of the region moves as a function of time. This dynamics model encompasses a wide variety of imaging scenarios. It does, however, ignore other factors influencing the dynamics of the images, such as variation in the illumination or specular reflections.

The terms $v_1(x_1, x_2, t)$ and $v_2(x_1, x_2, t)$ denote the components of the velocity vector field $\mathbf{v}(x_1, x_2, t) = [v_1(x_1, x_2, t),\ v_2(x_1, x_2, t)]^T$. Here, we use the bold lower case notation to indicate vectors. For the purposes of this thesis, we assume that the vector fields are linear in time. In other words, $\mathbf{v}(x_1, x_2, t) = \mathbf{v}(x_1, x_2)t$. This velocity vector field is sometimes called the optical flow field referring to the apparent image motion as opposed to the actual motion present in the 3-D *real-world* scene. Figure 1.1 shows a pair of images taken from the Yosemite Sequence of [14] and the motion vector field characterizing the image dynamics. The sequence simulates the measurements obtained while flying through the Yosemite valley. Here we immediately see the effect of perspective as the nearby valley wall in the lower left-hand portion of the sequence moves much faster than the ridges in the distance.

In general, the objective of motion estimation problems is that of estimating the vector field $\mathbf{v}(x_1, x_2)$, given measurements of the image sequence $f(x_1, x_2, t)$. In practice, we are given only sampled versions of the image sequence corrupted by measurement noise. As such, this task represents a challenging nonlinear estimation problem. For our purposes, we assume that the spatial sample spacing is T_x and the temporal sampling period is T_t reflective of the imaging system characteristics. For the remainder of the thesis, we will use the indices n_1, n_2 to refer to the discrete spatial sampling indices $f(n_1T_x, n_2T_x, kT_t)$, and refer to k as the temporal sampling index. To simplify the notation, we shall drop the sample periods T and use only n_1, n_2, k. Thus, the measurement model for the imaging system becomes

$$z(n_1, n_2, k) \quad = \quad f(n_1 - v_1(n_1, n_2)k, n_2 - v_2(n_1, n_2)k) + \epsilon(n_1, n_2, k) \tag{1.2}$$

The ϵ terms represent the additive measurement noise inherent to the imaging system. Such measurement noise represents a variety of sources such as image sensor thermal noise, stochastic randomness associated with photon arrivals, and electronic or readout noise. For the duration of this work, we model this random noise as being zero-mean, white (uncorrelated) Gaussian noise with variance or noise power σ^2. In practice, such a noise model has been found to accurately capture the effects of random noise in typical imaging systems [15].

In this thesis, we study several scenarios differing in the complexity of the motion vector field $\mathbf{v}(x_1, x_2)$. There are algorithms intended to estimate a completely arbitrary motion vector field such as [2]; however, in this thesis we focus on the class of vector fields which are parametric. The affine vector fields of interest are characterized by

$$\mathbf{v}(x_1, x_2) \quad = \quad \mathbf{v}_0 + \mathbf{M} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \tag{1.3}$$

where

$$\mathbf{v}_0 = \begin{bmatrix} v_{0_1} \\ v_{0_2} \end{bmatrix}, \tag{1.4}$$

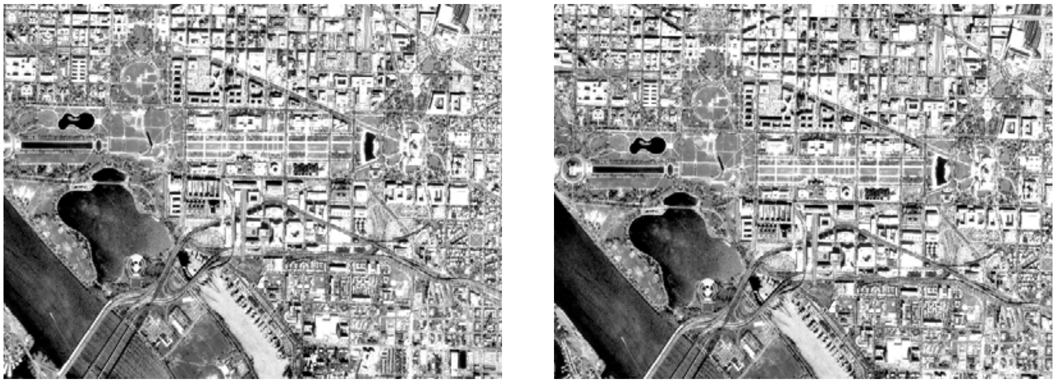

Figure 1.2: Example of global translational motion for the Washington DC satellite images.

is a constant vector representing global translational motion, and

$$\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \tag{1.5}$$

captures dynamics of rigid body motions as manifested in the image plane.

At first glance, such a restriction seems overly constricting given that a general image sequence may indeed contain highly complicated and arbitrary motion vector fields. As we show, however, such a model is applicable to a wide variety of scenarios. The utility of the affine motion model depends on the particular region of interest. For instance, the simple motion of global translation finds application in many scenarios where the imaging system is sufficiently far from a rigid object. A canonical example of such an imaging system is that of satellite imaging, where the apparent motion arises from relative motion between the satellite and the portion of earth under observation. Figure 1.2 shows a pair of satellite images which are related by a simple global translation. For such an imaging scenario, only the constant term $\mathbf{v}_0$ defining global translation is of interest. Within the context of our research, we refer to the problem of global parametric motion estimation between a pair of images as the problem of image registration.

Often times, only a portion of the scene under observation is of interest. In many scenarios, the full affine motion model works well to capture image dynamics produced by a stationary camera observing rigid object motion where the rigid object fills a significant portion of the camera's field of view. For example, Figure 1.3 shows a simple example of affine motion with respect to a moving book. As the book moves towards the camera, the motion vector field exhibits the effect of *zooming in*. The estimated motion vector field shown in Figure 1.3 was estimated from the cropped portion of the image containing the book.

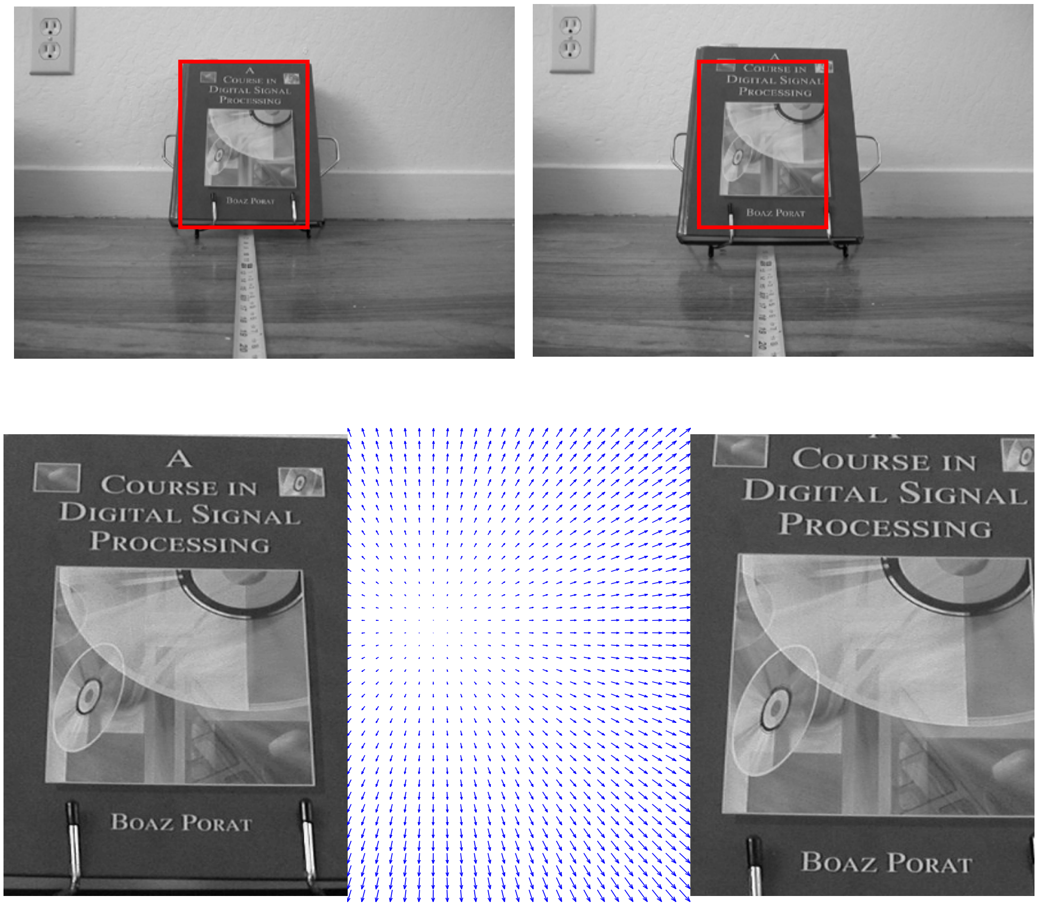

Figure 1.3: Example of affine motion.

In the previous example, we saw that within a local *window*, the motion was accurately captured by the affine motion model. As we shrink this window, the apparent motion becomes better modelled by simple translational motion. In fact, this observation forms the

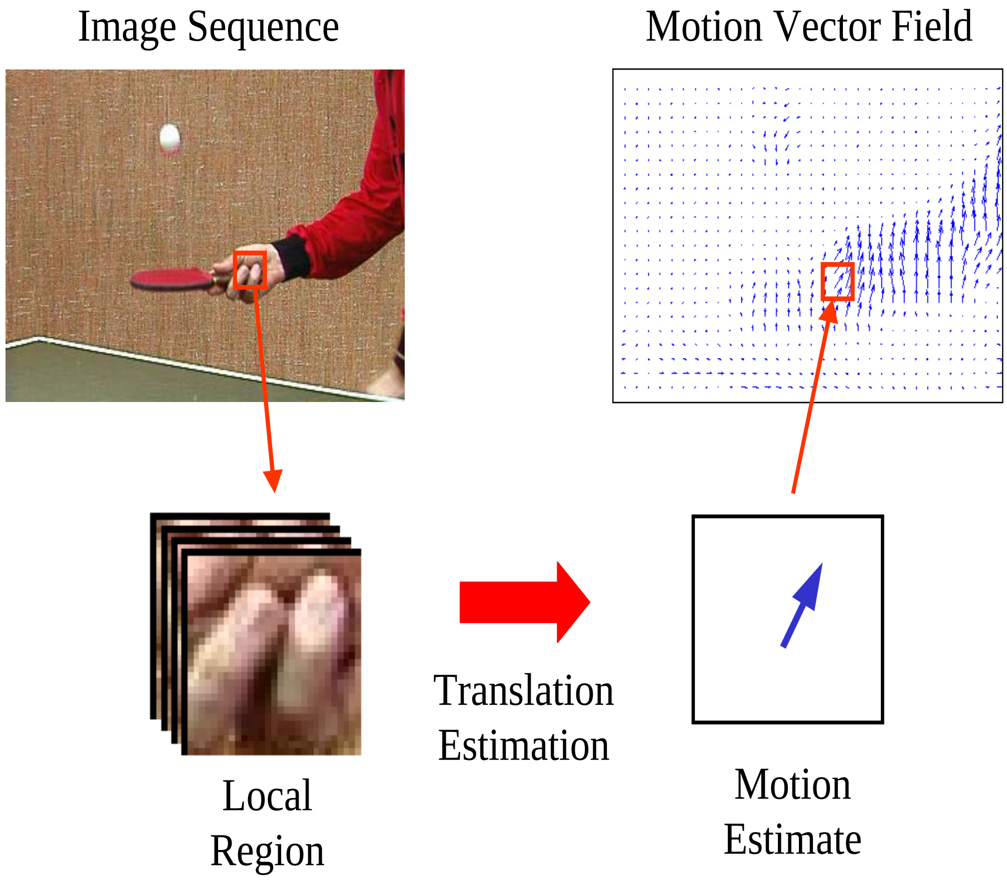

Figure 1.4: Example of local translation estimation.

basis for many motion estimation algorithms [14]. Figure 1.4 shows a typical example of local translational motion estimation. Finally, proper tiling of these local translational motion estimates can produce an overall estimate of very complicated motion vector fields $\mathbf{v}(x_1, x_2)$.

1.2 Applications of Motion Estimation

Motion estimation finds application in a very wide variety of research fields, each with its own specific operational characteristics, language, and methodology. In this section, we outline several research areas where motion estimation surface. In particular, we divide the applications of motion estimation into the two categories of image and scene analysis, and video processing, and compression.

Image and scene analysis, refers to those applications that focus on making inferences about the real-world scene under observation, using information related to the motion vector field. Traditionally, research related to such applications has arisen from the field of computer vision, a field whose foundations lie within the realm of computer science and specifically robotic vision. For example, in [9], the motion vector field is used to estimate the three-dimensional motion of the imaging system, sometimes referred to as ego-motion. Often the three-dimensional properties of rigid objects can be inferred from the motion vector field using what is known as structure from motion [4]. Other applications range from motion-based segmentation of objects in the scene [6] to human tracking and movement recognition [5]. In some sense, these applications are challenging in part because of the extremely varied operating scenarios. Much of the work is focused on completely arbitrary scenes imaged through video systems. As such, much of the development process has tended to utilize a qualitative or comparison-based performance evaluation. Traditionally, estimation theoretic analysis has been overlooked when examining these estimation problems. Most likely this is because of the general complexity of these applications and their origins in the computer science community.

Other applications where motion estimation plays a vital role fall into the category of video processing and compression. Many of these applications have traditionally been rooted in the field of signal processing. For instance, in video processing, motion estimation forms an essential component to most modern video compression algorithms [16], [10], [11]. Like computer vision applications, video compression applications make very few assumptions about the underlying video signals. In medical imaging, the sub-category of unimodal image registration is analogous to global motion estimation, often employing simple parametric models such as the affine motion model [17] as well as more general models [18]. Such registration is useful for diagnosing medical conditions and evaluation of medical procedures. The medical imaging scenario is distinct in that the types of images are constrained to be those of anatomical parts. Furthermore, medical imaging systems often have much poorer resolution and noise characteristics than optical systems.

Finally, we focus on the application of multi-frame image enhancement which originally motivated much of the research contained in this thesis. In multi-frame image enhancement, a set of images containing relative motions is fused to produce a single image of greater quality. Specifically, we focus on the problem of superresolution whereby the enhanced image is of greater resolution than the measured images. For instance, Figure 1.5 shows an example of superresolution using the robust algorithm described in [19]. In this example the rear of the vehicle is tracked through the sequence producing a set of images containing global affine motion. Using the estimates of these motion vector fields, a higher resolution image is reconstructed with less noise. It has been shown that motion estimation plays a critical role in the overall image enhancement performance of superresolution. Because of this, it is critical that the performance of motion estimation be well understood and characterized.

1.3 Contributions of the Dissertation

In this thesis we analyze the performance of motion estimation from the perspectives of computational efficiency and overall accuracy. We analyze the general problem from an estimation theoretic perspective, offering insight into the fundamental challenges and limitations associated with motion estimation. It is our goal that this work provide a common framework with which to evaluate and understand motion estimation performance. Hopefully, such a structure will form a bridge between the many fields using different forms and applications of motion estimation. To this end, this thesis offers contributions in two main areas. The first involves the analysis of the fundamental bounds limiting the accuracy of *any* estimation algorithm. The second contribution is a thorough analysis of the widely popular class of gradient-based algorithms and the proposal of a mechanism to greatly reduce their computational complexity.

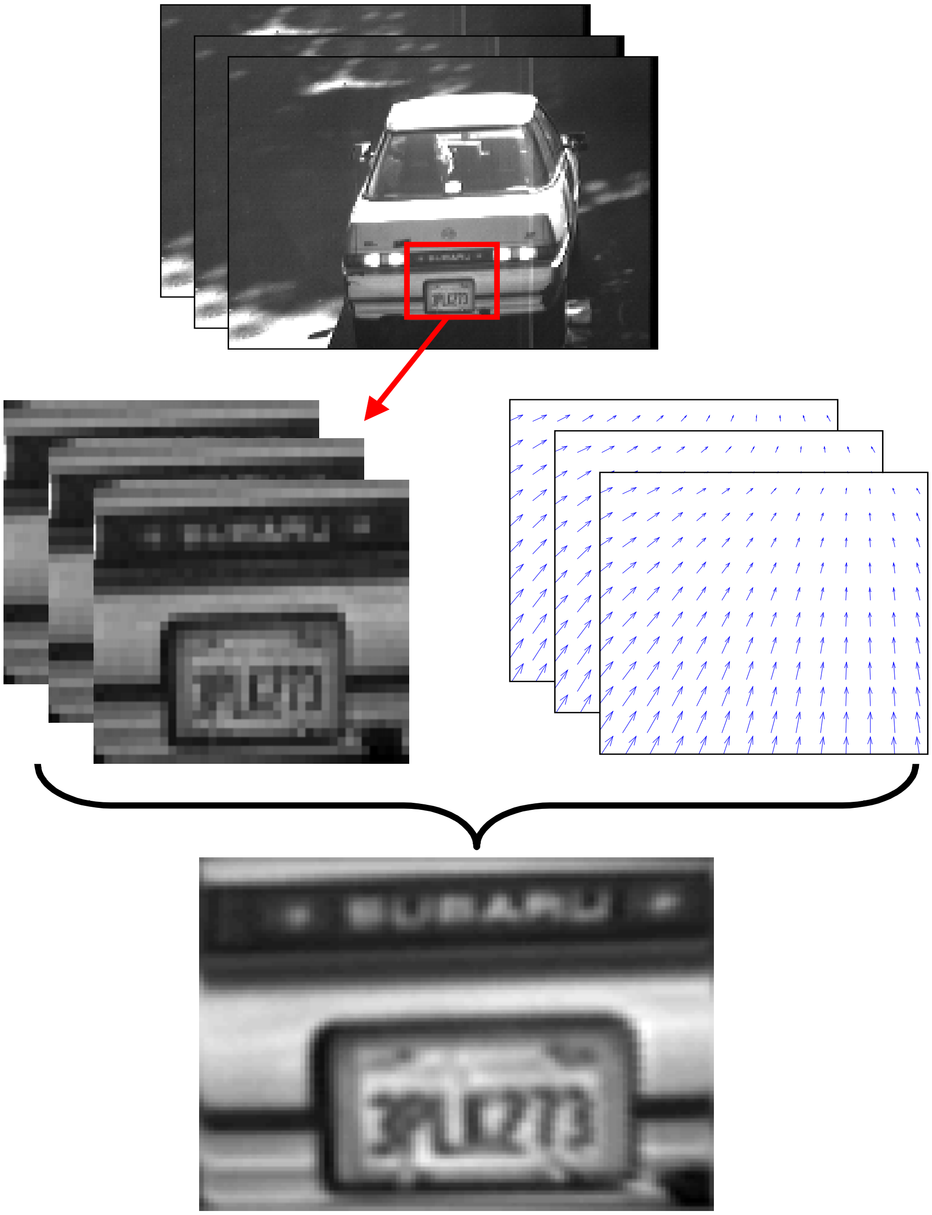

Figure 1.5: Example of superresolution enhancement.

- In Chapter 2, we propose using tomographic projections within the context of motion estimation as a vehicle to achieve dramatic computational speedups while sacrificing little in the way of estimator accuracy. Specifically, we explore the use of projections for gradient-based motion estimation.

- In Chapter 3, we bound the performance of a class of motion estimators using the Cramér-Rao bound, exploring the fundamental performance limits associated with translational motion estimation. In this chapter, we assume that the images are sampled above the Nyquist rate. We compare the performance of several popular algorithms with this bound, including the projection-based estimators proposed in Chapter 2.

- In Chapter 4, we characterize the bias for the class of gradient-based motion estimators. Using this bias formulation we construct rule-of-thumb performance limits for the gradient-based estimators. In addition, we suggest a novel method for improving estimator performance for low-noise scenarios where this estimator bias dictates performance.

- In Chapter 5, we extend our fundamental performance limits associated with translational motion estmiation to the sub-Nyquist (aliased) case, showing the implicit relationship between registration of aliased images and the problem of image reconstruction. We also show how such analysis relates to the problem of superresolution.

- In Chapter 6, we conclude the thesis and detail several possible directions for future work.

Chapter 2

Using Projections for Gradient-Based Motion Estimation

As we have shown, motion estimation represents a critical task for a variety of computer vision and video processing applications. Disparate as they may appear, these many applications share one common thread: in all such applications, the computational cost of performing accurate estimation of motion is very high, and this is often the bottleneck for both performance and real-time implementation. For instance, fast and accurate motion estimation is critical for any real-time motion compensating video encoder. In fact, most real-time video coders require special hardware to achieve the necessary motion estimation efficiency to support real-time encoding [13].

In this chapter, motivated by the need for fast and accurate motion estimation for compression, storage, and transmission of video, as well as for other applications, we present novel algorithms for estimating affine motion from video image sequences. Our methods utilize properties of the Radon transform to estimate image motion in a multiscale framework to achieve very accurate results. We develop statistical and computational models that motivate the use of such methods, and demonstrate that it is indeed possible to improve the computational burden of motion estimation by more than an order of magnitude, while maintaining the degree of accuracy afforded by the more direct, and less efficient, 2-D methods. We further demonstrate that multi-scale implementation of motion estimation algorithms using projections yields even more accurate and speedy estimates. The ability to improve computational complexity by almost an order of magnitude makes a compelling case for the routine use of projection-based methods in motion estimation [20–22].

2.1 Using Projections to Estimate Motion

The aim of this section is to show that a variety of motion estimation methods can be implemented in the Radon transform domain to yield very fast and accurate estimates of the motion parameters. The Radon transform (tomographic projection) of an image is defined as line integrals across the image [23]. It is well-known that pure translational motion in an image results in translation of the projections [23] along the direction of projection. This property has been used successfully in the past to estimate motion using projections [20–22, 24–32]. More recently, much of the (mostly ad-hoc) work in this area has been unified, producing a more general model of motion vector fields in the Radon transform domain [33] [34]. In particular, we show that affine motion in the image leads to affine motion in the projections as well[1]. We will use this property to derive efficient and accurate affine motion vector field estimators using projections.

2.1.1 Motion Under Projections

Before we begin the discussion of the use of projections in motion estimation, let us define the Radon transform. The Radon transform [23] of an image $f(x_1, x_2)$ is defined as

$$r(p,\phi) = \mathcal{R}_\phi\left[f(x_1,x_2)\right] = \int\int f(x_1,x_2)\delta(p - x_1\cos\phi - x_2\sin\phi)dx_1dx_2 \qquad (2.1)$$

where δ is the Dirac delta function. A projection of the image can be thought of as the Radon transform evaluated at a particular projection angle ϕ. As an example, Figure 2.1 shows a pair

[1] However, as we will elaborate later, the curl of the motion field is not directly measurable in the projections.

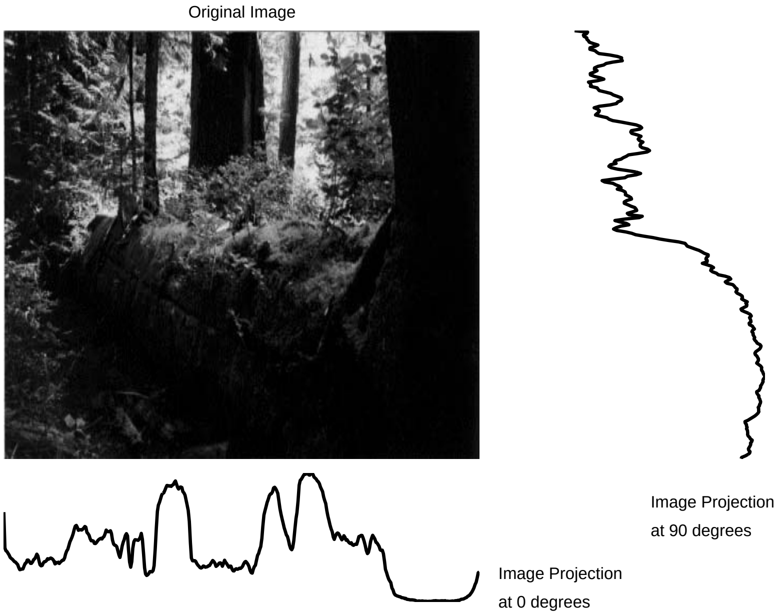

Figure 2.1: Set of tomographic projections of the forest image

of image projections at 0^o and 90^o. In this example, the projected image at 0^o represents the function created by summing all of the image intensity values in each column of the image. Similarly, the projection at 90^o represents the summation of each image row. In general, each point in the projection represents an integration along a line through the original image. From the definition we see that image projections are symmetric as $r(p, \phi) = r(-p, \phi + \pi)$. We note here that while the above definition represents the model for the Radon transform of a continuous image, in practice, we will use a discrete version of the Radon transform.

To understand how to estimate motion parameters indirectly using projections, we must first explore the relationship between motion in the original image sequence and the *induced* motion, or transformation in the projections. We begin our analysis for the simple case of translational motion, which is completely characterized by the shift vector $\mathbf{v}_0$. The simple relationship known as the shift property of the Radon transform [23] relates motion in images to the motion in projections by

$$\mathcal{R}_\phi \left[f(x_1 - v_{0_1}, x_2 - v_{0_2}) \right] = r(p - \mathbf{v}_0^T \mathbf{n}_\phi, \phi) = r(p - u_0(\phi), \phi), \tag{2.2}$$

where $\mathbf{n}_\phi = [\cos(\phi),\ \sin(\phi)]^T$ is a unit direction vector. Intuitively, each projection at angle ϕ "sees" the component of the vector $\mathbf{v}_0$ in the direction of the vector $\mathbf{n}_\phi$. Thus, a pure translation or shift given by $\mathbf{v}_0$ in the image domain results in a corresponding shift in the projection given by $u_0(\phi) = \mathbf{v}_0^T \mathbf{n}_\phi$.

The question of how general dynamics in image sequences behave under tomographic projection was addressed in [33], where it was shown that under certain smoothness conditions on the image function $f(x_1, x_2)$ and the vector field $\mathbf{v}(x_1, x_2)$, for sufficiently small Δt, there exists a unique function $u(p, \phi)$ such that

$$\mathcal{R}_\phi \left[f(x - v_1(x_1, x_2)\Delta t, y - v_2(x_1, x_2)\Delta t) \right] = r(p - u(p, \phi)\Delta t, \phi) \tag{2.3}$$

where

$$u(p, \phi) \frac{\partial r(p, \phi)}{\partial p} = \mathcal{R}_\phi [\mathbf{v}(x_1, x_2)^T \nabla f(x_1, x_2)] \tag{2.4}$$

and $\nabla f = [f_x(x_1, x_2),\ f_y(x_1, x_2)]^T$ denotes the spatial gradient of $f(x_1, x_2)$. As in [33], we refer to (2.4) as the Projected Motion Identity (PMI). This relationship suggests that for small transformations (where small depends on the product of the magnitude of the displacement vector field and the time elapsed Δt), the projections of a dynamic image sequence evolve in a qualitatively similar fashion as the original image sequence. That is, the projection function $r(p, \phi)$ evolves as a transformation or warping of the domain coordinates p by the function $u(p, \phi)$. It is important to note here that while the PMI is valid for *small* transformations of the image, it is more universally applicable when applied in a multiscale setting where at coarse scales, large warpings of the image are manifested as small transformations. We will elaborate on this point in a later section.

In the specific case of affine motion, it is shown in [33] that an affine motion vector field $\mathbf{v}(x_1, x_2)$ under projection behaves as

$$u(p, \phi) \approx \mathbf{v}_0^T \mathbf{n}_\phi + (\mathbf{n}_\phi^T \mathbf{M} \mathbf{n}_\phi)\, p = u_0(\phi) + \alpha(\phi)\, p. \tag{2.5}$$

This suggests that the projected motion $u(p, \phi)$ is also an affine function of the radial parameter p, and is parameterized by $u_0(\phi)$ and $\alpha(\phi)$. We note that the translational component of projected motion $u_0(\phi)$ depends only on the translational components of the original affine vector field, and the linear term $\alpha(\phi)$ depends only on the linear term in the original domain. This is part of a more general set of interesting properties of projected motion explored in detail in [33].

For the sake of completeness, it is worth mentioning that the exact form of the affine apparent motion in the projections is known and can be computed using properties of the Radon transform [23]. Namely, the exact form of the projected motion function is

$$u_{exact}(p,\phi) \quad = \quad \mathbf{v}_0^T \mathbf{n}_\phi + \left(1 - \frac{|\det(\mathbf{P})|}{\|\mathbf{P}^T \mathbf{n}_\phi\|_2}\right) p, \tag{2.6}$$

where $\mathbf{P} = \begin{bmatrix} 1 - m_{22} & m_{12} \\ m_{21} & 1 - m_{11} \end{bmatrix}$ satisfying $(\mathbf{I} - \mathbf{M})^{-1} = \frac{1}{|\det(\mathbf{P})|}\mathbf{P}$. Comparing (2.5) and (2.6), we observe that the only difference appears in the second term. Indeed, as is shown in Appendix 2.A, the term $\alpha(\phi)$ in (2.5) can be obtained by linearizing the term $\left(1 - \frac{|\det(\mathbf{P})|}{\|\mathbf{P}^T \mathbf{n}_\phi\|_2}\right)$ in (2.6) about $\mathbf{M} = \mathbf{0}$.

In any event, the exact form of the projected motion is highly nonlinear in the parameters of $\mathbf{M}$, and is not easy to use for motion estimation from projections. By contrast, in our approach, we estimate the affine parameters in a *linear* estimation framework. Employing this linear framework, as we will show, has the dual advantage of producing not only very fast but also quite accurate results.

It is instructive for the affine case to compare the exact formulation to the PMI formulation for a few specific cases:

1. **Pure Scaling -** For the case of pure scaling (e.g. zooming magnification) the affine parameters will have the form $\mathbf{M} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix}$. Using the exact form of the projected motion function we obtain

$$\begin{aligned} u_{exact}(p,\phi) = \left(1 - \frac{|\det(\mathbf{P})|}{\|\mathbf{P}^T \mathbf{n}_\phi\|_2}\right) p \quad &= \quad \left(1 - \frac{|1-s|^2}{(|1-s|)\|\mathbf{n}_\phi\|_2}\right) p \\ &= \quad (1 - |1-s|)p. \end{aligned}$$

On the other hand, using the linear form of (2.5) we obtain

$$u(p,\phi) \quad = \quad (\mathbf{n}_\phi^T M \mathbf{n}_\phi)\, p = s(\mathbf{n}_\phi^T \mathbf{n}_\phi)\, p = s\, p.$$

We observe that for scaling values of s less than 1, the two equations are equivalent.

2. **Pure Rotation -** For the case of pure rotation by angle ϑ the affine parameters will have the form $\mathbf{M} = \begin{bmatrix} 1-\cos\vartheta & -\sin\vartheta \\ \sin\vartheta & 1-\cos\vartheta \end{bmatrix}$. Thus, the exact form of the projected motion function is

   $$\begin{aligned} u_{exact}(p,\phi) = \left(1 - \frac{\det(\mathbf{P})}{\|\mathbf{P}^T\mathbf{n}_\phi\|_2}\right) p &= \left(1 - \frac{1}{\|\mathbf{n}_\phi\|_2}\right) p \\ &= 0. \end{aligned}$$

   This indicates that pure rotation, even in the exact formulation, conveys no information in a single projection. Meanwhile, the PMI approximation yields

   $$u(p,\phi) = (\mathbf{n}_\phi^T \mathbf{M} \mathbf{n}_\phi)\, p = (1-\cos\vartheta)\, p. \tag{2.7}$$

   Here we see that the approximation is close to the exact expression for small angles of rotation ϑ. We will again later elaborate on the difficulty of estimating rotation using projections and how this difficulty may be overcome.

2.1.2 Previous Work

The use of projections to estimate motion efficiently is not new. Very early works such as [24] use image projections at 0^o and 90^o to register translated images using a relative phase approach. More recently [22], and [26] have incorporated projections into correlation-based block motion estimators to speed up motion compensated video coding. In these works, the projections used to estimate translational motion were confined to 0^o and 90^o. Similarly, in [27] the authors use correlation between pairs of image projections at 0^o and 90^o to register translated images. Furthermore, they find that the use of projection effectively nullifies certain types of pattern noise, yielding improved performance over the direct methods. These works do not, however, address the question of estimating more general image dynamics such as affine motion.

A few researchers have utilized the Radon transform to estimate various forms of affine image motion. The authors of [28–30] use only a pair of image projections to accelerate motion detection and estimation of a subclass of affine motions, for use in video sequence processing and classification. They constrain the affine motion to that of global magnification and global translation to extract camera movement in video sequences. The work of [32] and [31] describes how the Radon transform could be used to estimate global rotation and translation in image sequences. In particular, [31] uses a set of 360 half image projections (approximately the set of projections at all angles) to accurately estimate global rotation and translation for manufacturing process control.

The above methods have not addressed the performance issues concerning the application of projections in estimating both global and local motion, particularly within a multiscale framework. The present work unifies most, if not all, of the above proposed approaches in a single framework, establishing a theoretical foundation for their use. In addition, the present work is the first to justify and use a gradient-based estimation scheme using projections based directly on the analysis of performance vs. computational complexity.

2.2 Gradient-Based Motion Estimation with Projections

In this section, we introduce the very accurate and widely-used class of motion estimation algorithms called the gradient-based algorithms. In particular, we propose a variant of the gradient-based motion estimation algorithm which utilizes tomographic projections to improve the computational efficiency of motion estimation.

2.2.1 Direct (2-D) Gradient-Based Affine Motion Estimation

The gradient based approach is a commonly used and effective method for directly estimating an optical flow field. Gradient based techniques or differential techniques compute image velocity directly from the image pixel intensities by expanding the right side of (1.1) in a Taylor series to obtain

$$f(x_1, x_2, t) = f(x_1, x_2, 0) - v_1(x_1, x_2)tf_{x_1} - v_2(x_1, x_2)tf_{x_2} + \dots$$

where $f_{x_1} = \frac{\partial}{\partial x_1} f(x_1, x_2, 0)$ represents the partial derivative of the image function with respect to x_1. Without loss of generality, we assume that we are examining a pair of images at times $t = 0, T_t$, and truncate the Taylor expansion to the first order thereby reducing this expression to the well known gradient constraint equation [14]

$$-f_t \quad = \quad \nabla f \cdot \mathbf{v}, \tag{2.8}$$

where $\nabla f = [f_{x_1},\ f_{x_2}]^T$ denotes the spatial gradient of f and f_t denotes the difference between two adjacent frames $f(x_1, x_2, T_t) - f(x_1, x_2, 0)$. Inserting the affine motion model (1.3) into (2.8), one obtains a linear equation in the unknown affine motion parameters:

$$\begin{aligned} -f_t \quad = \quad & v_{0_1}\ f_{x_1} + v_{0_2}\ f_{x_2} + m_{11}\ x_1\ f_{x_1} + \\ & m_{12}\ x_2\ f_{x_1} + m_{21}\ x_1\ f_{x_2} + m_{22}\ x_2\ f_{x_2}. \end{aligned} \tag{2.9}$$

This constraint can also arise from a more general assumption of intensity conservation where it is assumed that $df/dt = 0$, or the *total* derivative of the image brightness values does not change over some interval of time. Under this intensity conservation assumption, the model of (2.8) exactly characterizes the optical flow in the image sequence. Hence, f_t becomes the approximation of the partial derivative of the image sequence with respect to time.

In general, the spatio-temporal gradients must be approximated from the given image data using derivative filters

$$\tilde{f}_{x_1} \quad \approx \quad g_1(n_1, n_2) * *z(n_1, n_2, 0) \tag{2.10}$$

$$\tilde{f}_{x_2} \quad \approx \quad g_2(n_1, n_2) * *z(n_1, n_2, 0) \tag{2.11}$$

where $**$ represents a 2-D convolution operation with the gradient filters g_1 and g_2. Typically, these gradient filters are chosen to be short finite impulse response (FIR) filters which are finite approximations to the ideal infinite impulse response (IIR) derivative filters. We will revisit these choice of these filters in Chapter 3.

This motion model of (2.9) is assumed to apply to a spatiotemporal region of the image sequence represented by Ω. Thus, over the region Ω (which may in fact be the entire image) we obtain a linear system of equations of the form

$$-\mathbf{z} = \mathbf{A}\mathbf{\Phi} + \mathbf{e}. \tag{2.12}$$

Here, $\mathbf{z}$ denotes the vector whose elements are each pixel differences $z(n_1, n_2, 1) - z(n_1, n_2, 0)$ in the region Ω scanned in some particular fashion (e.g. raster-scanned). $\mathbf{e}$ represents noise or other departures from the assumed model. The vector $\mathbf{\Phi}$ is the vector of unknown motion parameters defining the motion vector field in the region Ω, as in

$$\mathbf{\Phi} \;\; = \;\; [v_{0_1} \; v_{0_2} \; m_{11} \; m_{12} \; m_{21} \; m_{22}]^T \, . \tag{2.13}$$

Finally, the matrix $\mathbf{A}$ contains the terms of (2.9) where the spatial gradients have been approximated using (2.10) and (2.11). In other words, the rows of $\mathbf{A}$ are given by:

$$\left[\tilde{f}_{x_1} \; \tilde{f}_{x_2} \; x_1\tilde{f}_{x_1} \; x_2\tilde{f}_{x_1} \; x_1\tilde{f}_{x_2} \; x_2\tilde{f}_{x_2}.\right] \tag{2.14}$$

Each row vector corresponds to a pixel location in the region Ω scanned in a fashion similar to $\mathbf{z}$.

Typically, it is assumed that the noise term $\mathbf{e}$ is zero-mean Gaussian noise. Under this assumption, the best (minimum variance) linear, unbiased estimate of the parameters of interest is given by the least-squares approach [35]:

$$\begin{aligned} \hat{\mathbf{\Phi}} \;\; &= \;\; -\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{z}, && (2.15) \\ \text{Cov}(\hat{\mathbf{\Phi}}) \;\; &= \;\; (\mathbf{A}^T\mathbf{A})^{-1}. && (2.16) \end{aligned}$$

At times, it is appropriate to associate different weights with the pixels in the region Ω. For example, it is common to apply a weighting function which *focuses* the estimator on the center of a block. Such weighting takes the form of a diagonal matrix $\mathbf{W}$ where the elements along the diagonal are the weights associated with a particular pixel. When the weighting is applied the estimator becomes

$$\begin{aligned} \hat{\boldsymbol{\Phi}} &= -\left(\mathbf{A}^T\mathbf{W}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{W}\mathbf{z}, && (2.17) \\ \text{Cov}(\hat{\boldsymbol{\Phi}}) &= (\mathbf{A}^T\mathbf{W}\mathbf{A})^{-1}. && (2.18) \end{aligned}$$

In practice, even for a reasonably small region (5×5 pixels), the gradient-based estimator usually provides quite accurate estimates of the affine parameters of the vector field $\mathbf{v}$. The performance of this method and its variations has been studied at some depth in [36–38]. The work of [37] originally outlined the methods for estimating optical flow in a global parametric framework, describing both the models used in this chapter for the global translational and global affine model and other more complicated models. In [36], the authors propose a region-based optical flow estimation scheme where the blocks are assumed to contain affine motion. Furthermore, the work of [38] explores the use of robust estimators within the context of gradient-based optical flow estimation. While the methods contained in these articles achieve high degrees of accuracy, the computational complexity of the methods is often quite high. The purpose of this chapter is to introduce motion estimation using tomographic projections. As we will show, the use of tomographic projections can be incorporated into a variety of motion estimation schemes to achieve substantial speedup with little or no loss in performance. Specifically, we explore the use of projections in gradient-based motion estimation.

2.2.2 Estimating Projected Motion Parameters

Earlier, we showed that the motion in the projections, or the projected motion, is accurately characterized by the projected motion function $u(p, \phi)$ which, in turn, is parameterized by $u_0(\phi)$ and $\alpha(\phi)$. We now present a method for estimating the projected motion parameters $u_0(\phi)$ and $\alpha(\phi)$ from projections at a fixed angle ϕ over time based on a one-dimensional analog of the gradient-based method.

As we did in the derivation of the direct gradient-based estimator, we expand the right side of (2.3) in a Taylor series

$$r(p,\phi,t) = r(p - u(p,\phi)t, \phi) = r(p,\phi) + r_p(p,\phi)u(p,\phi) + \dots .$$

Ignoring the higher order terms, we obtain

$$-r_t(p,\phi) = r_p(p,\phi)\ u(p,\phi). \tag{2.19}$$

where r_p denotes the partial derivatives of $r(p,\phi,t)$[2] with respect to the location variable p and $r_t = r(p,\phi,t) - r(p,\phi,0)$. Interestingly, a corollary of the result (2.3), proved in [33], is that if the intensity conservation assumption $df/dt = 0$ is invoked in the image domain, the corresponding constraint holds in the projection domain: $dr/dt = 0$. As before, this assumption implies that the model of (2.19) exactly describes the relationship between image derivatives and image motion. Again, in the context of this assumption r_t refers to the partial derivative of the projected image sequence with respect to time.

Similar to the 2-D case, inserting the affine model (2.5) into (2.19) we obtain

$$-r_t = u_0(\phi)\ r_p + \alpha(\phi)\ r_p\ p$$

As in the direct method, we assume the motion model applies over the projection of the region Ω which we denote Ω_p. Note that we refer to the projection of $z(n_1, n_2, k)$ at an angle ϕ as $z_p(n_p, \phi, k)$. The subscript p refers to data or functions in the projected domain.

As with the 2-D case, gathering the measurements over the region Ω_p we generate an overdetermined system of linear equations

$$-\mathbf{z}_p(\phi) = \mathbf{A}_p(\phi)\boldsymbol{\Phi}_p(\phi) + \mathbf{e}_p(\phi) \tag{2.20}$$

where $\mathbf{z}_p(\phi)$ is the a vector containing the projection pixel difference $z_p(n_p, \phi, T_t) - z_p(n_p, \phi, 0)$ for $n_p \in \Omega_p$ at a particular angle ϕ. The vector $\boldsymbol{\Psi}_p$ is the vector of unknown projected motion parameters

$$\boldsymbol{\Phi}_p(\phi) \quad = \quad [u_0(\phi)\ \alpha(\phi)]^T$$

[2] We note here that $r(p,\phi,t)$ is the Radon transform of $f(x_1, x_2, t)$ for each fixed t

Finally, the rows of the matrix $\mathbf{A}_p$ are given by

$$[\tilde{r}_p \; p\tilde{r}_p]$$

measured at every location in Ω_p. Here, the approximation of the partial derivatives $\tilde{r}_p(p)$ is done in a special fashion that takes into account the geometry of the image region. The discussion of this calculation is presented in Appendix 2.B. It is worth noting here an interesting relationship between the noise $\mathbf{e}$ in the image domain formulation of (2.12) and the noise $\mathbf{e}_p(\phi)$ in the corresponding projection domain (2.20). The noise term $\mathbf{e}_p(\phi)$ is a projection of the random field $\mathbf{e}$, and as such will still be assumed to be zero-mean. However, assuming the random field comprising the error term $\mathbf{e}$ to be white, with variance σ^2, the noise vector $\mathbf{e}_p(\phi)$ will have a diagonal covariance matrix $\mathbf{C}_\phi = \sigma^2 diag[S^{-1}(\phi)]$, where the function $S(\phi)$ reflects the geometry of the random field region (See Appendix 2.B for further details).

Thus, solving equation (2.20) in a weighted least squares sense we obtain:

$$\begin{aligned} \hat{\mathbf{\Phi}}_p(\phi) &= -(\mathbf{A}_p^T \mathbf{C}_\phi^{-1} \mathbf{A}_p)^{-1} \mathbf{A}_p^T \mathbf{C}_\phi^{-1} \mathbf{z}_p(\phi) && (2.21) \\ \text{Cov}(\hat{\mathbf{\Phi}}_p(\phi)) &= (\mathbf{A}_p^T \mathbf{C}_\phi^{-1} \mathbf{A}_p)^{-1} && (2.22) \end{aligned}$$

As before, if we choose to apply an additional weighting function to the data within Ω_p, captured by the diagonal matrix $\mathbf{W}_p$, the weighted estimates of the projected motion parameters become

$$\begin{aligned} \hat{\mathbf{\Phi}}_p(\phi) &= -(\mathbf{A}_p^T \mathbf{W}_p \mathbf{C}_\phi^{-1} \mathbf{A}_p)^{-1} \mathbf{A}_p^T \mathbf{W}_p \mathbf{C}_\phi^{-1} \mathbf{z}_p(\phi) && (2.23) \\ \text{Cov}(\hat{\mathbf{\Phi}}_p(\phi)) &= (\mathbf{A}_p^T \mathbf{W}_p \mathbf{C}_\phi^{-1} \mathbf{A}_p)^{-1} && (2.24) \end{aligned}$$

The covariance terms of (2.22) and (2.24) are 2×2 matrices of the form

$$\begin{bmatrix} C_{u_0,u_0}(\phi) & C_{u_0,\alpha}(\phi) \\ C_{\alpha,u_0}(\phi) & C_{\alpha,\alpha}(\phi) \end{bmatrix} \qquad (2.25)$$

2.2.3 Estimating Motion Parameters From Projected Motion Parameters

Having just described the method for estimating the motion parameters in the Radon transform domain in the previous section, we are now in a position to present the final step in estimating the parameters of the original 2-D motion model. Namely, the model (2.5), which relates affine motion in the image domain to the motion in projections can now be invoked. By comparing terms on the left and right-hand sides of (2.5), we can directly observe that

$$\begin{aligned} u_0(\phi) &= \mathbf{n}_\phi^T \mathbf{v}_0, \\ \alpha(\phi) &= \mathbf{n}_\phi^T \mathbf{M} \mathbf{n}_\phi \end{aligned}$$

This pair of identities allows the estimation of parameters of both the translational part $\mathbf{v}_0$ and the purely linear part $\mathbf{M}$ of the vector field $\mathbf{v}(x_1, x_2)$.

Assuming the projected motion parameters have been estimated as $\hat{u_0}(\phi)$ and $\hat{\alpha}(\phi)$ at a set of angles ϕ_i, $i = 1, \cdots, N_\phi$, we can collect all such estimates and write

$$\begin{bmatrix} \hat{u}_0(\phi_1) \\ \vdots \\ \hat{u}_0(\phi_{N_\phi}) \end{bmatrix} = \begin{bmatrix} \cos\phi_1 & \sin\phi_1 \\ \vdots & \vdots \\ \cos\phi_{N_\phi} & \sin\phi_{N_\phi} \end{bmatrix} \mathbf{v}_0 + \epsilon_0,$$

$$\begin{bmatrix} \hat{\alpha}(\phi_1) \\ \vdots \\ \hat{\alpha}(\phi_{N_\phi}) \end{bmatrix} = \begin{bmatrix} \cos^2\phi_1 & \sin^2\phi_1 & 2\cos\phi_1 \sin\phi_1 \\ \vdots & \vdots & \vdots \\ \cos^2\phi_{N_\phi} & \sin^2\phi_{N_\phi} & 2\cos\phi_{N_\phi} \sin\phi_{N_\phi} \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{22} \\ m_{12} + m_{21} \end{bmatrix} + \epsilon_\alpha,$$

or equivalently,

$$\begin{aligned} \mathbf{y}_0 &= \mathbf{R}_0 \mathbf{v}_0 + \epsilon_0 \qquad (2.26) \\ \mathbf{y}_\alpha &= \mathbf{R}_\alpha \mathbf{m} + \epsilon_\alpha \end{aligned}$$

Because the noise terms ϵ_0 and ϵ_α are in general correlated, we combine these estimates into one system of the form

$$\begin{bmatrix} \mathbf{y}_0 \\ \mathbf{y}_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_\alpha \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{m} \end{bmatrix} + \begin{bmatrix} \epsilon_0 \\ \epsilon_\alpha \end{bmatrix} \quad or \quad \mathbf{y} = \mathbf{R}\Phi_r + \epsilon$$

where the subscript r indicates the reduced set of 2-D affine motion parameters. Here, we use the subscript r to reflect that we are estimating a *reduced* set of 2-D motion parameters due to the inability to estimate the curl component.

The error vector ϵ is assumed to be zero-mean with a banded covariance matrix $\mathbf{C}_\epsilon$. The covariance matrix $\mathbf{C}_\epsilon$ is constructed from the collection of covariance matrices $Cov(\Phi_p(\phi))$ of (2.25). Then the matrix $\mathbf{C}_\epsilon$ is constructed as

$$\mathbf{C}_\epsilon = \begin{pmatrix} C_{u_0,u_0}(\phi_1) & 0 & 0 & C_{u_0,\alpha}(\phi_1) & 0 & 0 \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & C_{u_0,u_0}(\phi_{N_\phi}) & 0 & 0 & C_{u_0,u_0}(\phi_{N_\phi}) \\ C_{\alpha,u_0}(\phi_1) & 0 & 0 & C_{\alpha,\alpha}(\phi_1) & 0 & \\ 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & C_{\alpha,u_0}(\phi_{N_\phi}) & 0 & 0 & C_{\alpha,\alpha}(\phi_{N_\phi}) \end{pmatrix}$$

Finally, we estimate $\mathbf{\Phi_r}$ via weighted least squares:

$$\hat{\mathbf{\Phi}}_r = (\mathbf{R}^T\mathbf{C}_\epsilon^{-1}\mathbf{R})^{-1}\mathbf{R}^T\mathbf{C}_\epsilon^{-1}\mathbf{y} \tag{2.27}$$

When estimating only the translational component of motion, the forward model reduces to (2.26). The covariance matrix for ϵ_0, namely $\mathbf{C}_{\epsilon_0}$, is a diagonal matrix whose terms are given by $C_{u_0}(\phi_i)$; the final estimate of the 2-D translation parameters is given by

$$\hat{\mathbf{v}}_0 = (\mathbf{R}_0^T\mathbf{C}_{\epsilon_0}^{-1}\mathbf{R}_0)^{-1}\mathbf{R}_0^T\mathbf{C}_{\epsilon_0}^{-1}\mathbf{y}_0. \tag{2.28}$$

Ultimately, we will compare the performance of these projection-based estimators with that of the original 2-D estimation methods.

2.2.4 Vector Field Curl under Projections

It is important to recall that a drawback of using a projection-based estimator is the inability to directly estimate all of the parameters of $\mathbf{M}$ uniquely. Namely, we cannot estimate the component $m_{12} - m_{21}$ under projection. While the $m_{12} + m_{21}$ term represents a measure of the shearing of the image sequence, the missing term $m_{12} - m_{21}$ corresponds to the curl of the motion vector field. As we indicated earlier, this suggests that pure rotation will not be distinguishable in a single projection even in the case of the exact projected affine model of (2.6). At first glance, it would appear that estimating rotational motion is then not at all possible from projections; however, this is not the case. Indeed, if the complete set of projections of the images were computed, then the angle of rotation could be easily determined by computing pairwise correlation coefficients between a projection (at, say, $\phi = \phi_0$) and the many other available projections. The angle of rotation is then determined by the difference in the projection angles of the pair of projections with highest spatial correlation coefficient. In our method, in order to keep the computational complexity to a minimum, we deal with only a small number of projections (3 or 4) sampled sparsely in the range $[0, \pi]$; therefore, the correlation approach is impractical.

Fortunately, our method can still be modified and employed to estimate purely rotational motion. Though we do not pursue this specific problem in this chapter, we shall indicate how this can be done by recalling an important property of projected motion. It was proved in [33], and mentioned earlier in this chapter, that projected motion satisfies the linearity property so that translational motion maps to a single component u_0 in the projections and the linear part $\mathbf{M}$ maps to another separate component $\alpha(\phi)$ in the projections. This linearity idea can be further exploited to show that the complementary rotational and irrotational components of motion also are separated in the projections. The implication here is that if we simply ignore the fact there is a rotational component in the vector field of interest, or equivalently, if we assume that $m_{12} - m_{21} = 0$, then the resulting estimated motion vector field is *purely* irrotational. With this fact in mind, given an arbitrary affine motion vector field, we can proceed by first estimating the irrotational component according to the projection-based approach described above. The images then can be warped according to this estimated vector field, and the resulting pair of images will then be *known* to be related by a vector field that is a combination of translational and purely rotational components. While the rotation can not be estimated using a global application of the projection-based method, it is possible to estimate rotation by applying the method locally in smaller windows of the image. It is true that in a window of fixed size, as we move away from the center of rotation the curl component becomes increasingly small. Therefore, the component of pure rotation in a window away from the center of rotation is measured effectively as a translation. Combining these local estimates with the knowledge that the underlying motion field is *purely* rotational with an unknown center of rotation (the translational component), the curl component of the overall global vector field very likely can be accurately estimated as well. Of course, the computational complexity of the overall projection-based method process is worsened if this additional rotational motion estimation is carried out. We leave further analysis of this problem for future research.

In the present framework, in order to generate estimates for all of the affine parameters, we assume that $m_{12} - m_{21} = \rho$ where ρ is some known curl value, typically set to zero. In closing this section, it is also worth observing that we need at least two projection angles to determine the shift vector $\mathbf{v}_0$ and at least three projection directions to estimate all of the curl-free affine parameters of $\mathbf{M}$. Given an arbitrary affine vector field, we typically employ four projection angles at $\phi = 0, 45, 90,$ and 135 degrees. The choice of these angles can also be optimized as a function of the given image (spatial frequency) content to produce the best possible estimates – this is another interesting topic worthy of future research.

2.2.5 Global, Local, and Multiscale Estimation

Until now, we have not specified the region of interest where we apply the above estimators. In this section we explain how the previously described models can be applied to the image sequence in a global or local fashion to estimate more complex vector fields. Then, we show how the estimators can be embedded into a hierarchical or multiscale framework to yield improved performance as well as computational efficiency.

In earlier sections, estimators (2.27) and (2.15) were applied to an unspecified region in the image Ω where the affine motion model was assumed to characterize the image dynamics.

The simplest such region to apply the estimator is the entire image. For this case, we obtain parameters that describe the global motion. When the motion model applies in the global sense, this form of estimation usually produces a very good estimate as often there are thousands of equations used to estimate only six parameters.

Another popular approach for estimating more complex vector fields is that of dividing the images into small overlapping or non-overlapping regions. This region-based approach assumes that the simple parametric model characterizes the motion present only in a small region. The more complex vector field $\mathbf{v}(x_1, x_2)$ is then approximated as a piecewise collection of simpler parametric vector fields. These piecewise vector fields are sometimes forced to satisfy some constraint such as smoothness [39]. The simplest form of local estimation is to find translational motion for small image regions. The translational model of image dynamics $f(x_1 - v_{0_1}t, x_2 - v_{0_2}t)$ is likely to be valid for small spatio-temporal regions in the image sequence. The vector field estimation process begins by estimating the translational motion for each region in the image. Then, these estimates are combined to generate an estimate of the vector field $\mathbf{v}(n_1, n_2)$. The estimated translational motion for each block represents a sample of the overall vector field. Thus, the dense vector field estimate $\hat{\mathbf{v}}(n_1, n_2)$ is usually generated by some form of interpolation of these vector field samples. One such form of interpolation is the replication of the vector samples, where the final vector flow field has regions of constant velocity such as in Figure 2.2. This approach is common in video coding where the motions of each block are estimated using a variety of approaches. Some of these approaches include matched filtering, correlation and phase-based methods.

As shown in [40], this local vector field estimation method can be understood as a special case of variable sized region-based motion estimation. Multiscale motion estimation attempts to estimate a vector field by estimating the velocity components for variable sized regions at different scales of image resolution. Basically, the multiscale framework estimates a vector field by combining the coarse motion properties in large image regions at low image resolution with the finer motion vector estimates estimated in smaller regions at higher resolution.

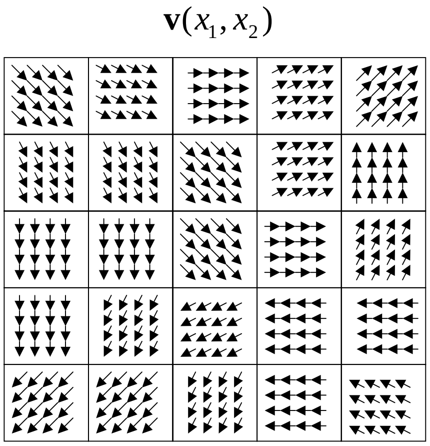

Figure 2.2: Region Based Vector Field Estimation

To understand the utility of the multiscale framework, we first describe the iterative estimation process. Recall from Sections 2.2.1 and 2.2.2 the truncation of the Taylor series expansion to the first order used to produce (2.8) and (2.19). This approximation assumes a small motion vector $\mathbf{v}(x_1, x_2)$ (assuming unit time between frames) and is not accurate for regions where the velocity vector $\mathbf{v}(x_1, x_2)$ is large. The multiscale approach attempts to remedy this inaccuracy by iterating over scale. More specifically, the multiscale approach decomposes the image sequence into a dyadic pyramid of successive sequences of lowpass filtered and downsampled images, as shown in Figure 2.3. Such multiscale decomposition is applied to each frame in the video sequence. This creates an image sequence pyramid with sequences at the top having the lowest resolution and size while the original sequence lies at the bottom. The motion vectors describing the dynamics in the downsampled images will necessarily be reduced by a factor of 2 at every level of the pyramid. This reduction in magnitude improves the accuracies of the models (2.8) and (2.19) by "shrinking" the magnitude of $\mathbf{v}(x_1, x_2)$. Furthermore, it has been shown that the lowpass filtering used to construct the image pyramid also serves to regularize the optical flow estimation problem [40].

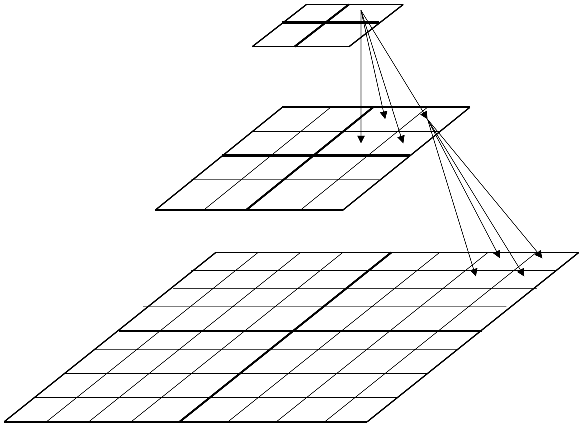

Figure 2.3: Dyadic pyramid used in multiscale estimation.

When the assumption of intensity conservation is violated in an image sequence, the estimates produced by by (2.15) and (2.27) contain errors. These errors partially result from modelling errors arising from the linearization of a nonlinear problem. One generic method to mitigate these errors is to use multiple estimation iterations in a Gauss-Newton type scheme [37]. In general, the performance of the iterative nonlinear least squares estimators depend on both the convexity of the objective function (sum of the squared image differences) as well as the accuracy of the relative estimate at each iteration.

An iterative nonlinear least squares estimation can be combined with the multiscale framework. The iterative multiscale estimation begins by estimating motion in the image sequence at the coarsest scale (the top of the pyramid), working in a coarse-to-fine strategy using the 2-D estimator (2.15) or the projection-based estimation (2.27) at each level of the pyramid. The image sequence at a particular level of the pyramid is denoted by $z^l(n_1, n_2, k)$ where the superscript of z indicates the level of the pyramid where L is the total height of the pyramid. Each level of the pyramid is constructed by first filtering the sequence by a low-pass filter $h(n_1, n_2)$ followed by a downsampling operation by a factor of 2. In other words

$$z^l(n_1, n_2, k) \quad = \quad \Big[h(n_1, n_2) ** z^{l-1}(n_1, n_2, k)\Big] \downarrow_2 \qquad (2.29)$$

where $\downarrow_2$ represents a downsampling by a factor of 2. Because of the image downsampling, the velocity vector field $\mathbf{v}^l(n_1, n_2)$ for a given level of the pyramid is reduced in magnitude by a factor of 2^l from the original $\mathbf{v}(n_1, n_2)$. Initially, the vector field $\hat{\mathbf{v}}^L(n_1, n_2)$ is estimated from the image sequence $z^L(n_1, n_2, k)$ at the coarsest level. Secondly, the image sequence at the next finer resolution level of the pyramid $z^{L-1}(n_1, n_2, k)$ is warped according to twice the velocity estimates $2\hat{\mathbf{v}}^L(n_1, n_2)$, creating a warped image sequence $\check{z}^{L-1}(n_1, n_2, k)$ with the estimated coarse image motion *removed* from the image sequence. Finally, the residual motion $\hat{\mathbf{v}}^r(n_1, n_2)$ is estimated from this warped image sequence yielding an updated velocity vector field estimate given by

$$\hat{\mathbf{v}}^{L-1}(n_1, n_2) = 2\hat{\mathbf{v}}^L(n_1, n_2) + \hat{\mathbf{v}}^r(n_1, n_2). \tag{2.30}$$

This process repeats down the pyramid iterating in a coarse to fine fashion. The multiscale aspect of the iteration serves the additional role of reducing computation since the images at the coarsest levels are downsampled (smaller). Thus, the computation time required to warp the image sequences as well as the time required to estimate the residual motions is reduced.

The multiscale iteration can be applied to both the direct and the projection based method for estimating vector fields. Using of multiscale iteration for direct estimation has been shown to produce very accurate results [37]. The multiscale iteration can also be combined with projection based estimation to produce equally good results while realizing significant computational savings. For example, Figure 2.4 shows the Fake Trees image at the coarsest resolution ($L = 3$) and at the original image resolution. The corresponding image projections also are shown and are used to estimate global motion. Initially, the global motion parameters are estimated from a set of projections of the coarsest image sequence. The process proceeds as detailed, only at every step a projection-based motion estimation algorithm is employed. In Section 2.3, we present experiments showing the performances of the multiscale methods relative to the non-iterative methods.

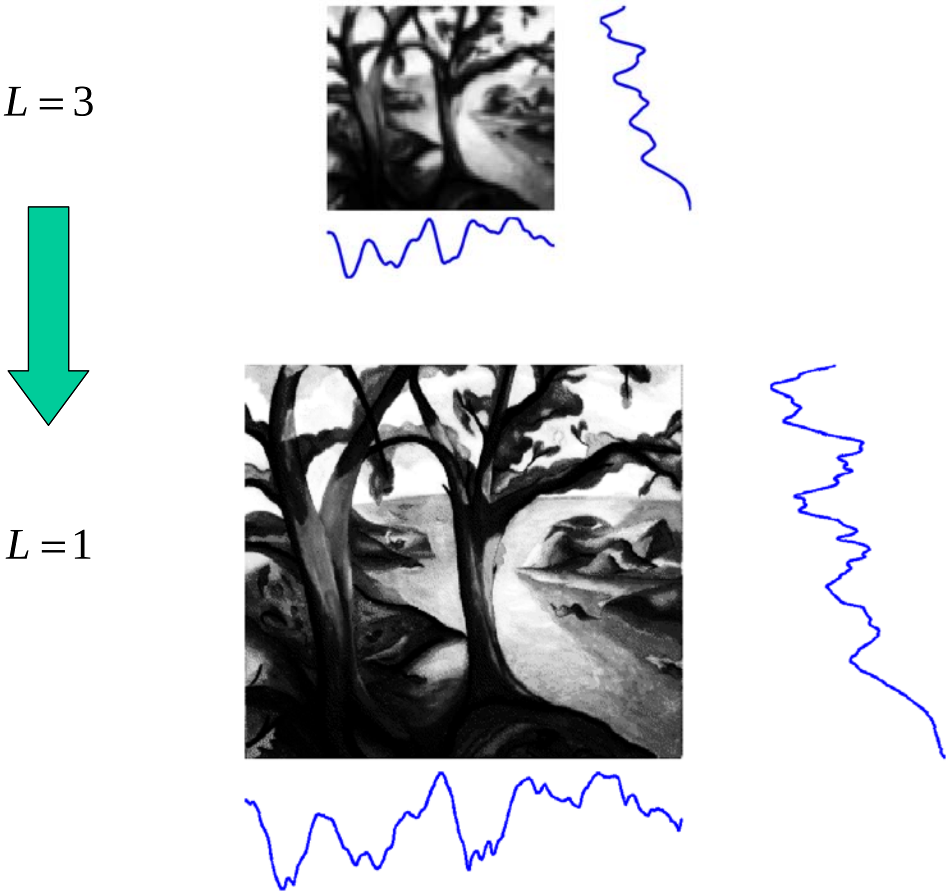

Figure 2.4: Fake Trees image at two pyramid resolutions and the corresponding projections.

Gradient-Based Estimators	2-D	1-D
Projection	0	$N_\phi N^2$
Gradient	$10N^2$	$5N_\phi N$
Motion Estimation	$36N^2$	$4N_\phi N$
Inverse Estimation	0	$36N_\phi^2$

Table 2.1: Complexity of Gradient-Based Direct and Indirect Methods

2.2.6 Computational Complexity

In this section we compare the computational complexities of the direct and the projection-based estimators for estimating global motion. We will examine the computational cost of estimating the parameters of affine motion between a pair of $N \times N$ images (without a loss of generality we assume that the images are square). We are not including any of the cost associated with multiscale estimation as it will pertain to both estimators equally. We distinguish the original estimator from the projection based estimator as being the 2-D and 1-D methods respectively. We assume that N_ϕ is the number of projections used (typically 3 or 4). For our evaluation of image gradients, we use convolution kernels such that 10 multiplications and additions are required to estimate the 2-D gradient at each pixel and 5 multiplications and additions are required to estimate the derivative at each point in the projection. We obtain the cost for motion estimation as a general cost of solving a linear system from [41] where six parameters are estimated in the 2-D case and two are estimated in the 1-D case. Finally, we assume that $N_\phi \ll N$ so that the final cost of estimating the 2-D affine parameters from projected motion parameters is negligible. This leads us to a general overall computational complexity of $\mathcal{O}(46N^2)$ for the direct 2-D estimation and $\mathcal{O}(N_\phi N^2 + 9N_\phi)$ for the projection-based 1-D estimator. We find in practice that using $N_\phi = 4$ projection angles to estimate affine motion requires at worst only about 25 percent of the computational time required by the 2-D method, thus realizing significant computational savings. It is important to note that the cost of computing projections, which is the leading term in the complexity of the 1-D method, involves only additions, while the leading N^2 term in the direct 2-D method involves multiplications. Furthermore, we point out that many motion estimation methods typically employ some form of presmoothing of the images prior to motion estimation. We have not included this presmoothing step in our analysis or experiments and we have ignored its computational cost. But we mention here that the computational cost of presmoothing is again significantly lower if this operation is performed on the projections instead of on the images.

2.3 Experiments

We present a set of experiments exploring the performance of the direct and indirect (projection-based) methods for estimating affine motion. We begin with experiments estimating global affine vector fields for a set of images in both a non-iterative and multiscale iterative framework. Then, we compare the direct and indirect estimation of general vector fields using local estimation methods. For our experiments, we use a combination of well-known benchmark image sequences as well as our own synthesized image sequences.

2.3.1 Error Measures and Test Image Sequences

Following [14], we measure mean angular error between the correct motion vector field $\mathbf{v}(n_1, n_2)$ and the estimated motion vector field $\hat{\mathbf{v}}(n_1, n_2)$. In keeping with the method of [14], we utilize two difference performance measures. The first is called the mean angular error (MAE). To compute the MAE, we write the 2-D vector field as a 3-D vector function over a 2-D scalar field as

$$\mathbf{V}(n_1, n_2) \quad = \quad [v_1(n_1, n_2), v_2(n_1, n_2), 1]^T$$

where v_1, v_2 are the velocities in the 2 spatial dimensions. The mean angular error between $\mathbf{V}(n_1, n_2)$ and $\widehat{\mathbf{V}}(n_1, n_2)$ is measured by:

$$MAE = \frac{1}{N^2} \sum_{n_1, n_2} \arccos\left(\frac{\mathbf{V}(n_1, n_2)^T \widehat{\mathbf{V}}(n_1, n_2)}{\|\mathbf{V}(n_1, n_2)\|_2 \ \|\widehat{\mathbf{V}}(n_1, n_2\|_2} \right) \tag{2.31}$$

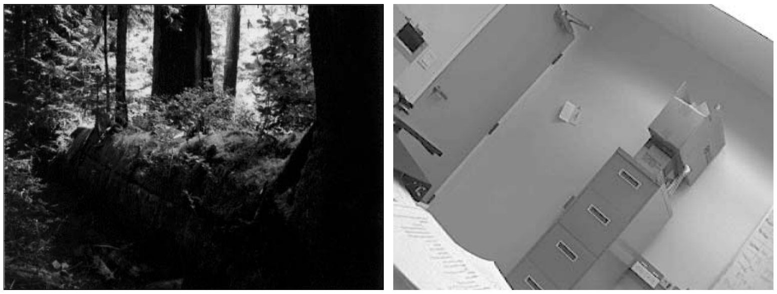

Figure 2.5: Experimental Test Images: Forest (left) and Lab (right)

where the sum is taken over all N^2 pixels of interest. To gather more information about the motion estimation performance, we also compute the mean magnitude error (MME) as:

$$MME = \frac{1}{N^2} \sum_{n_1,n_2} \|\mathbf{v}(n_1,n_2) - \hat{\mathbf{v}}(n_1,n_2)\|_2 \tag{2.32}$$

Again, this represents the average magnitude of the error vector over all pixels in the image.

In our experiments, we evaluate the performance of our projection-based estimator both for well-known image sequences and for our own synthetic image sequences. To generate a synthetic image sequence, we warp an individual image according to the affine transformation model of (1.3) to create an image pair. The second image in the pair is a linearly interpolated version of the reference image, where the interpolation is based on a known motion vector field. We then estimate this vector field from the image pair. The images we used to generate synthetic image sequences are shown in Figure 2.5.

1. **Forest -** Picture of a forest containing similar image statistics to those of a natural scene with rich textures. The image is 300×440 pixels.

2. **Lab -** Picture from a webcam at the researchers' office. The webcam was rotated about 45^o so as to create an image in which the majority of image texture is not aligned at 0^o and 90^o. The image is 240×320 pixels.

In addition to our own synthetic image sequences, we measure performance on a well known set of benchmark image sequences from [14]. While these image sequences contain many frames, we limit the image sequences to only 5 frames. In practice, this represents a reasonable number of frames; in real image sequences the vector field $\mathbf{v}$ often remains static for only a short period of time. The image sequences that we use are the following:

1. **Diverging tree -** The image sequence imitates a camera zooming into scene creating a divergent motion vector field.

2. **Translating tree -** The image sequence contains mostly global translational motion arising from camera motion in the x_1-direction. The translational motion vectors are approximately 2.5 pixels per frame.

3. **Yosemite -** The image sequence contains a more complex motion field from perspective effects of an imaging system flying through Yosemite valley. A sample of the image sequence and corresponding motion vector field is shown in Figure 1.1.

Both the Translating and Diverging tree sequences are based on the image shown in Figure 2.12. We apply both the global and local estimators to these benchmark sequences.

For each set of global estimation experiments we add zero-mean Gaussian noise to produce the specified image signal to noise ratio [3](SNR). The motion vector fields were estimated from these noisy image sequences and the corresponding error measures for the estimates were calculated. For each experiment, we repeated the estimation process 100 times at each SNR and averaged both the MAE and MME performance measures. Figure 2.6 shows an example of the Tree image at different SNRs.

We evaluate the performance of the local estimation methods without adding noise to the sequences to allow a comparison of our results with those of [14].

2.3.2 Global Affine Estimation

We begin our experimental performance analysis by estimating global affine vector fields described by the affine motion model of (1.3). As mentioned in Section 2.2.4, the

[3] Signal to noise ratio (SNR) is defined as $10 \log_{10} \frac{\sigma_c^2}{\sigma^2}$ where σ_c^2 and σ^2 are the variances of a clean frame and the noise respectively.

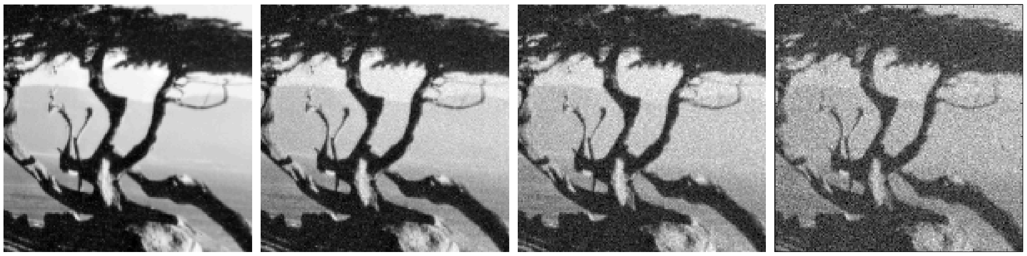

Figure 2.6: Tree image at different SNR values (70db, 30db, 15db, 10db)

rotational component of the affine vector field cannot be directly estimated using the global projection-based estimator. Therefore, we first examine the performance of the method in estimating affine vector fields constrained to have no rotational component, and compare the results to the performance of the direct 2-D method[4]. We then extend the experiments to include estimation of the general affine model to understand the indirect estimator's performance in the presence of image rotation. For the projection-based estimation, we use four projection angles of $0^o, 45^o, 90^o$ and 135^o in each experiment.

We initially examine the performance of the projection based global estimator on the benchmark Translating and Diverging Tree sequences, which contain no rotational component.

The plots of Figures 2.7 and 2.8 show the performance of the 1-D and 2-D methods using no multiscale iteration ($L = 1$, dashed lines) and for a multiscale pyramid of height $L = 3$ (solid lines). The triangles indicate the error of the 2-D method and the circles indicate the error of the 1-D projection based estimator. We follow this graphical format for all of the experiments on global affine vector field estimation.

From Figures 2.7 and 2.8, we see that the projection-based estimator *outperforms* the direct 2-D method when the method is not iterated in multiscale, but the difference in performance shrinks as the SNR improves. In both image sequences, when motion is estimated using

[4] In the interest of fairness, the 2-D method employed in estimating these irrotational vector fields employed *constrained* least squares with the constraint that $m_{12} - m_{21} = 0$. The plots of Figures 2.7, 2.8, and 2.9 reflect the use of this constraint in the 2-D case.

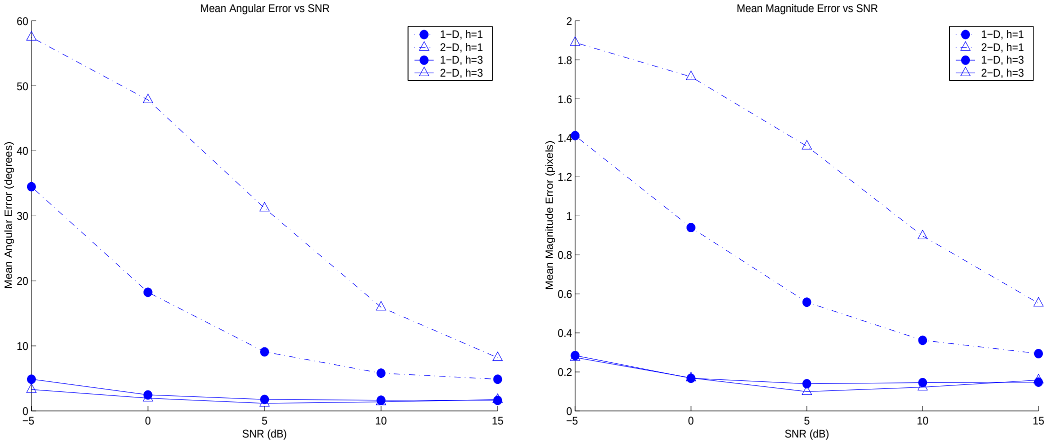

Figure 2.7: Mean Angular and Magnitude Error for the Translating Tree sequence

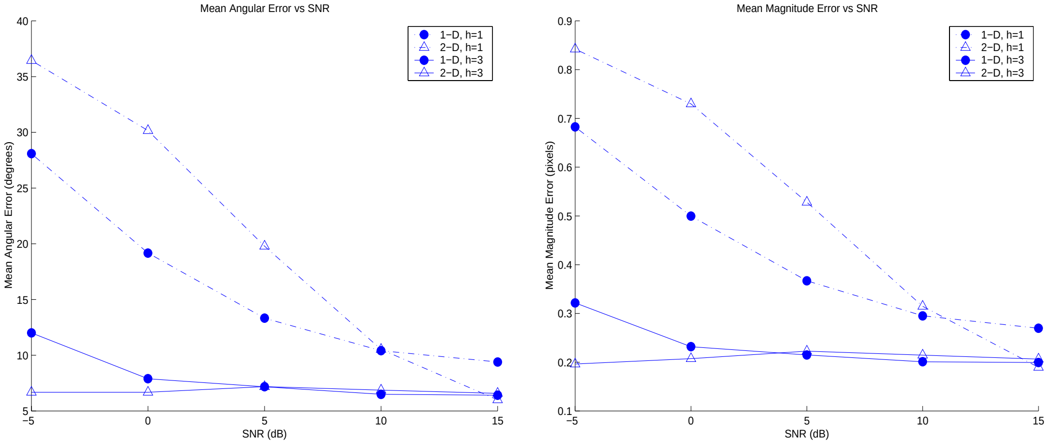

Figure 2.8: Mean Angular and Magnitude Error for the Diverging Tree sequence

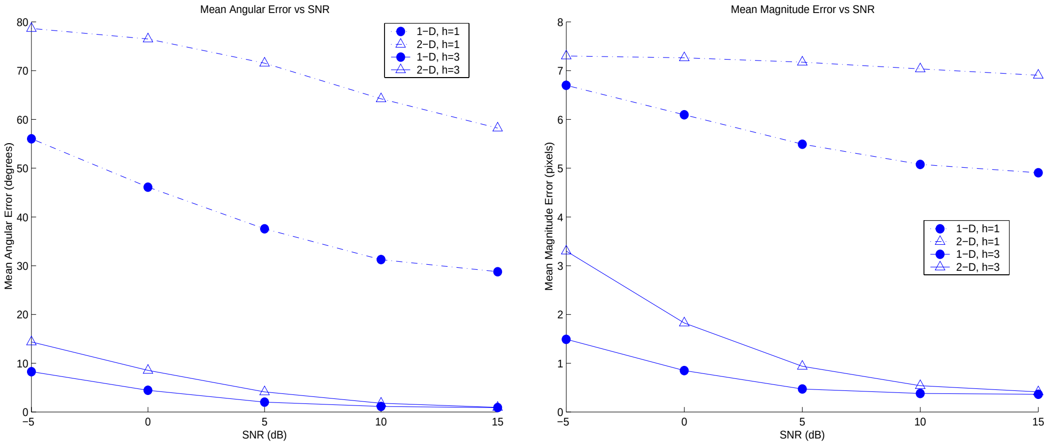

Figure 2.9: Mean Angular and Magnitude Error for the Forest image with constrained motion

multiscale iteration, the performance of the direct and projection based estimators are essentially equivalent. Only for very poor SNR in the case of the Diverging Tree sequence (Figure 2.8) do we see a small performance difference between the 1-D and 2-D methods.

To evaluate the performance of the projection-based estimator more systematically using simulated motion, we continue our experimentation using our synthetic image sequences. Figure 2.9 shows the performance of both the 2-D and 1-D methods in estimating the global affine vector field with parameters $\mathbf{M} = \begin{bmatrix} .05 & .01 \\ .01 & .06 \end{bmatrix}$ and $\mathbf{v}_0 = [.5, \ .5]^T$ applied to the Forest image.

As a point of reference, for a particular realization of noise at SNR of 5 dB, the 1-D estimator using multiscale ($L = 3$) iteration produces estimates of $\widehat{\mathbf{M}} = \begin{bmatrix} .0484 & .0079 \\ .0079 & .0382 \end{bmatrix}$ and $\hat{\mathbf{v}}_0 = [.3223, \ .4986]^T$ which corresponds to mean angular error of 1.8 degrees and a mean magnitude error of 0.39 pixels. Using the same data, the 2-D estimator produces $\widehat{\mathbf{M}} = \begin{bmatrix} .0471 & .0080 \\ .0080 & .0339 \end{bmatrix}$ and $\hat{\mathbf{v}}_0 = [.3885, \ .1760]$ which corresponds to a mean angular error of 3.19 degrees and a mean magnitude error of 0.68 pixels.

Again, we see the non-iterative projection-based estimator outperforming the direct

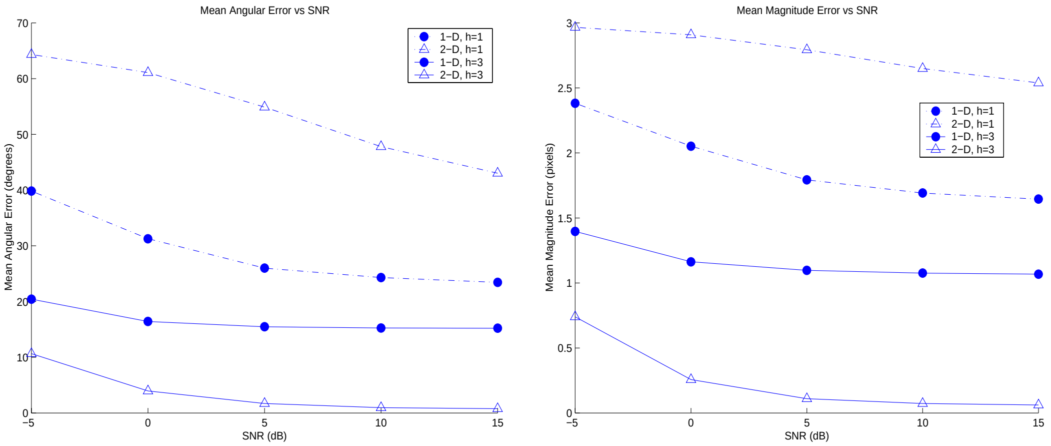

Figure 2.10: Mean Angular and Magnitude Errors for the Lab image with rotation

2-D estimator. Using the multiscale iteration, the 1-D projection based estimator continues to outperform the 2-D method. As the SNR improves, both methods seem to converge to similar performance. We present these results as a representative sample of the many experiments we carried out using other irrotational affine vector fields as well as different reference images.

To analyze the performance for the case of general affine motion, we estimate image dynamics for a vector field containing nonzero curl. Figure 2.10 shows the errors in estimating a vector field applied to the Lab image with affine parameters $M = \begin{bmatrix} -.01 & -.01 \\ -.03 & .02 \end{bmatrix}$ and $\mathbf{v}_0 = [.5,\ .5]^T$.

As the plot indicates, without using multiscale iteration, the projection-based 1-D estimator seems to outperform the 2-D estimator. Presumably, the 1-D method is more robust when estimating gross motions than the 2-D method. However, when employing a multiscale pyramid of height $L = 3$, the 2-D method clearly produces better estimates of the vector field. While the multiscale iteration does improve the projection-based estimates, the iterations only improve the estimate of the irrotational component of motion. For example, at a SNR of 5 dB and multiscale height $L = 3$, the projection-based method produces affine parameter estimates

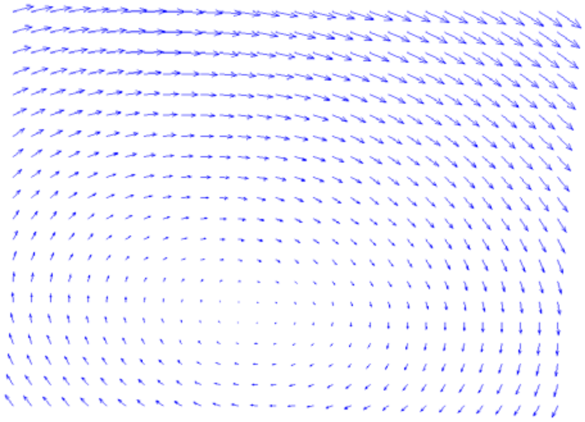

Figure 2.11: Residual velocity vector field for projection-based estimation of general affine vector field.

of $\widehat{\mathbf{M}} = \begin{bmatrix} -.0093 & -.0238 \\ -.0238 & .0178 \end{bmatrix}$ and $\hat{\mathbf{v}}_0 = [-.6807,\ .0944]^T$. The residual motion vector field $\mathbf{v} - \hat{\mathbf{v}}$ is shown in Figure 2.11. This figure shows that the residual motion not captured by the projection-based estimator is primarily the rotational component of affine motion. By contrast, the 2-D estimator for the same image pair produces the estimates $\widehat{\mathbf{M}} = \begin{bmatrix} -.0106 & -.0097 \\ -.0294 & .0188 \end{bmatrix}$ and $\hat{\mathbf{v}}_0 = [-.5291,\ .4231]^T$, effectively estimating the curl of the vector field.

These experiments indicate that when the motion is constrained such that there is no image rotation, the 1-D method performs just as well if not better than the 2-D method for global affine motion estimation. Even when rotation was present, the 1-D method appears to offer more robust estimation in the presence of large scale motion as evidenced the performance differences for the non-multiscale estimation. The notion that the 1-D method can perform better than the 2-D method in some circumstances deserves systematic and careful future study. The previous figures also show that the multiscale iteration can provide substantial improvements in performance for both the non-iterative 1-D and 2-D estimators.

2.3.3 Local Translation Estimation

Finally, we present experiments with the use of projections for estimating local motion in a block-based scheme as outlined in Section 2.2.5. As mentioned earlier, application of the direct gradient-based translational estimation of Section 2.2.1 to small blocks in an image sequence was first introduced by Lucas and Kanade [42]. Here, we compare the performance of a projection-based block-wise translational estimation scheme with the direct 2-D gradient-based method of [42]. The direct gradient method consistently performs well as shown in most optical flow estimation survey papers such as [14] and [13]. We will show that this performance also extends to the projection-based method, while significantly improving the computational efficiency.

As indicated in Section 2.2.5, both the direct and indirect techniques require choosing a set of operating parameters, ultimately affecting estimator performance. For instance, both methods initially subdivide the image into blocks for which a motion vector is estimated. The choice of block sizes plays a critical role in determining both the accuracy and the speed of the techniques. Furthermore, depending on a desired density of the motion vector field, the size of the blocks affects the amount of block overlap. Both methods must choose a number of images to use in calculating one motion vector field. Finally, each of the projection-based approaches requires a pair of projection angles.

To improve the performance of the block based estimators, we apply a weighting vector to the least squares estimator which weights the pixels at the center of the block more than the pixels at the periphery. We denote this weighting function $w(x_1, x_2)$ for the direct estimator and $w(p)$ for the indirect estimator. Applying this weighting function to larger blocks will maximize localization accuracy while minimizing the risk of an ill-conditioned system of equations. Basically, the weighting function forces the estimator to estimate motion primarily from the pixels at the center of the block, but also allows pixels at the periphery of the block to influence the estimate slightly. To simplify the characterization of the weighting function, we use Gaussian functions $w(x_1, x_2) \approx e^{-\frac{x_1^2+x_2^2}{\varrho}}$ and $w(p) \approx e^{-\frac{p^2}{\varrho}}$. The weighting function is

Method	1-D Tran	2-D Tran	1-D Div	2-D Div	1-D Yos	2-D Yos
MAE (degrees)	11.385	14.108	5.888	6.112	18.820	21.195
Std	0.7064	0.6470	0.3325	0.3361	0.7245	0.7861
MME (pixels)	0.574	0.778	0.153	0.169	1.120	1.023
Std	0.0269	0.0231	0.0094	0.0110	0.0503	0.0359
Cpu Time (s)	1.920	23.880	1.930	24.030	7.530	96.160

Table 2.2: Results for Translating Tree, Diverging Tree, and Yosemite

parameterized by ϱ, or the variance of the Gaussian function.

To directly compare the 1-D block based estimator with the 2-D block based method in a fashion similar to [14], we estimate the general motion vector fields for the Translating and Diverging Tree and the Yosemite sequences using overlapping blocks of size 30×30 pixels which appears to produce the best overall results for both methods. The width of the Gaussian functions was $\rho = 6$ which suggests that the majority of the estimator weight is placed within the center 5 pixels or so. We then use both estimators on each sequence using 5 frames and tabulated the results in Table 2.2. The same table also includes the computation time required to estimate the vector fields.

From Table 2.2, we observe that the accuracy of the 1-D and 2-D methods appear to be statistically equivalent. The computational complexity, however, is dramatically reduced in the projection-based approaches. The 1-D method's total computation time was on average about 90 percent better than the 2-D counterpart. As a visual example, Figure 2.12 shows the estimated motion vector fields for the Diverging Tree image sequence overlaid atop one image of the sequence. Note that the motion vector fields are visually quite similar. As one might expect, the performance in estimating a globally affine vector using a local method is inferior to that of estimating the parameters in a global fashion. The poor performance can be explained by the sensitivity of local models to large motions. For example, the magnitude of the motion vectors in the Translating tree sequence is about 2-3 pixels per frame, explaining the performance degradation using local estimation. Similarly, portions of the Yosemite sequence contain very large motions which are very difficult to estimate in a local fashion. Much of

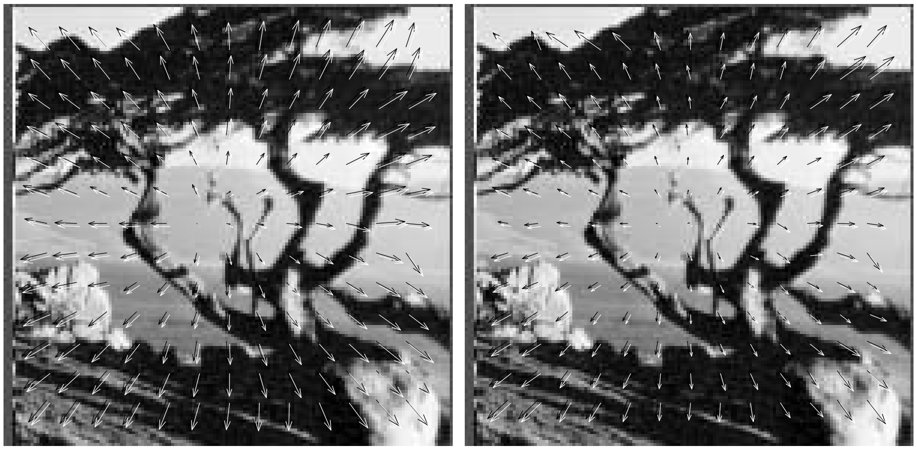

Figure 2.12: Estimated motion vector field superimposed on the image using 2-D (left) and 1-D (right) estimators for the Diverging tree sequence.

the motion in the Diverging tree sequence is sub-pixel, explaining the significantly improved performance on this sequence.

2.4 Conclusion

In this chapter we introduced a unified framework for the estimation of affine motion parameters using tomographic projections. Previous attempts at the same were mostly ad-hoc and, most importantly, did not address the question of relative performance between the direct 2-D methods and the proposed 1-D approaches. Here we have shown that projection-based methods offer a computationally attractive alternative to the direct methods, while in most cases maintaining or even improving the level of accuracy. The idea that projection-based methods often can display improved performance is theoretically intriguing and deserves careful study in the future. In Chapter 4, we compare the performance of projection-based and the standard direct gradient-based algorithms for estimating translation. Such analysis offers some insight into the observed improved estimator performance associated with the projection-based algorithms. We have also shown that the projection-based method can be combined with a multiscale iterative framework to provide further accuracy in motion estimation while minimizing computation time.

These results suggest much room for future research in the area of estimating motion using projections. For instance, the gradient-based method is only one of many methods for estimating motion using projections. Phase-based methods are another possibility that should be explored [43]. Improved performance may also be realized by using more sophisticated statistically robust methods in place of the least squares approach presented in this chapter. Finally, some of our preliminary experimentation has indicated that the choice of projection angles plays a fundamentally important role in the performance of any projection-based motion estimation method. Adaptively identifying the optimal set of projection angles, as a function of the given images, for best estimator performance remains an open question.

2.A Linearized Projected Affine Motion

In this appendix, we derive the Maclaurin series approximation of the exact form of the projected motion function $u(p, \phi)$ for affine motion. From (2.6) we see that the exact form of the affine motion under projection is

$$u_{exact}(p, \phi) = \mathbf{v}_0^T \mathbf{n}_\phi + \left(1 - \frac{|det(\mathbf{P})|}{\|\mathbf{P}^T \mathbf{n}_\phi\|_2}\right) p \tag{2.33}$$

We show how the coefficient of the second term in the above expression can be linearized by expanding it in a first order Maclaurin series. To begin, let us define

$$\alpha_{exact}(\mathbf{P}) \quad = \quad 1 - \frac{|det(\mathbf{P})|}{\|\mathbf{P}^T \mathbf{n}_\phi\|_2}. \tag{2.34}$$

Next, we rewrite (2.34) as a function of the four affine parameters as follows

$$\begin{aligned}
\alpha_{exact}(\mathbf{P}) \quad &= \quad \alpha_{exact}(m_{11}, m_{12}, m_{21}, m_{22}) \\
&= \quad 1 - \frac{|1 - m_{11} - m_{22} + m_{11}m_{22} - m_{12}m_{21}|}{[((1 - m_{22})\cos(\phi) + m_{21}\sin(\phi))^2 + (m_{12}\cos(\phi) + (1 - m_{11})\sin(\phi))^2]^{1/2}}
\end{aligned}$$

The first order Maclaurin series of $\alpha(\mathbf{P})$ will have the form

$$\alpha_{exact}(\mathbf{P}) \quad = \quad \alpha(\mathbf{I}) + m_{11}\frac{\partial\alpha(\mathbf{I})}{\partial m_{11}} + m_{12}\frac{\partial\alpha(\mathbf{I})}{\partial m_{12}} + m_{21}\frac{\partial\alpha(\mathbf{I})}{\partial m_{21}} + m_{22}\frac{\partial\alpha(\mathbf{I})}{\partial m_{22}} \tag{2.35}$$

To simplify the derivation, we write

$$\alpha_{exact}(\mathbf{P}) = 1 - \beta(\mathbf{P})\zeta^{-1/2}(\mathbf{P})$$

where

$$\beta(\mathbf{P}) = |1 - m_{11} - m_{22} + m_{11}m_{22} - m_{12}m_{21}|$$

and

$$\zeta(\mathbf{P}) = ((1 - m_{22})\cos(\phi) + m_{21}\sin(\phi))^2 + (m_{12}\cos(\phi) + (1 - m_{11})\sin(\phi))^2 \qquad (2.36)$$

Thus, from the chain rule we see that the partial derivatives of α_{exact} will have the form

$$\alpha_x = -\left[\frac{\partial\beta}{\partial x}(\zeta^{-1/2}) - (\beta\frac{1}{2}\zeta^{-3/2})\frac{\partial\zeta}{\partial x}\right] = \left[(\beta\frac{1}{2}\zeta^{-3/2})\frac{\partial\zeta}{\partial x} - \frac{\partial\beta}{\partial x}(\zeta^{-1/2})\right].$$

Next, we note that $\alpha_{exact}(\mathbf{0}) = 0$, $\zeta(\mathbf{0}) = 1$ and $\beta(\mathbf{0}) = 1$.

We now compute the partial derivatives of β evaluated at 0.

$$\begin{aligned}
\beta_{11}(\mathbf{0}) &= -1 \\
\beta_{12}(\mathbf{0}) &= 0 \\
\beta_{21}(\mathbf{0}) &= 0 \\
\beta_{22}(\mathbf{0}) &= -1
\end{aligned}$$

Likewise, we now evaluate the partial derivatives of ζ.

$$\begin{aligned}
\zeta_{11}(\mathbf{0}) &= -2\sin^2(\phi) \\
\zeta_{12}(\mathbf{0}) &= 2\cos(\phi)\sin(\phi) \\
\zeta_{21}(\mathbf{0}) &= 2\cos(\phi)\sin(\phi) \\
\zeta_{22}(\mathbf{0}) &= -2\cos^2(\phi)
\end{aligned}$$

Finally, we see that the partial derivatives of α_{exact} are

$$\begin{aligned}\alpha_{11}(\mathbf{0}) &= 1-\sin^2(\phi)=\cos^2(\phi)\\ \alpha_{12}(\mathbf{0}) &= \cos(\phi)\sin(\phi)\\ \alpha_{21}(\mathbf{0}) &= \cos(\phi)\sin(\phi)\\ \alpha_{22}(\mathbf{0}) &= 1-\cos^2(\phi)=\sin^2(\phi)\end{aligned}$$

Combining these calculations, we obtain the following linearization of α_{exact}:

$$\begin{aligned}\alpha_{exact}(\mathbf{M}) &\approx m_{11}\cos^2(\phi)+m_{12}\cos(\phi)\sin(\phi)+m_{21}\cos(\phi)\sin(\phi)+m_{22}\sin^2(\phi)\\ &= \mathbf{n}_\phi^T\mathbf{M}\mathbf{n}_\phi \qquad (2.37)\end{aligned}$$

This is the same form of projected affine motion obtained using the PMI assumption, discussed in (2.5).

2.B Calculating Derivatives in Image Projections

Here we will introduce the intuitive reasoning for applying a weighting to the projection images prior to calculating derivatives used in estimating projected motion. We shall explain how this weighting acts as a modification of the spatial derivative operator. Because the image under projection is defined over a rectangular region of samples, different points in the projection are generated by integrating over lines of varying length. In terms of image pixels, this means that different points in the projection integrate different numbers of pixels in the original image. Thus, a rectangular constant valued image on $[-\frac{X_1}{2},\frac{X_1}{2}]\times[-\frac{X_2}{2},\frac{X_2}{2}]$ would not appear flat in the projection image but rather as a piecewise linear function (see Figure 2.13) given by

$$\begin{aligned}\mathcal{R}[f(x_1,x_2)=c] &= \int_{-\frac{X_2}{2}}^{\frac{X_2}{2}}\int_{-\frac{X_1}{2}}^{\frac{X_1}{2}} c\,\delta(p-x_1\cos(\phi)-x_2\sin(\phi))dx_1dx_2\\ &= \int_{S^-(p,\phi)}^{S^+(p,\phi)} cds\\ &= S^+(p,\phi)-S^-(p,\phi)=S(\ \phi)\end{aligned}$$

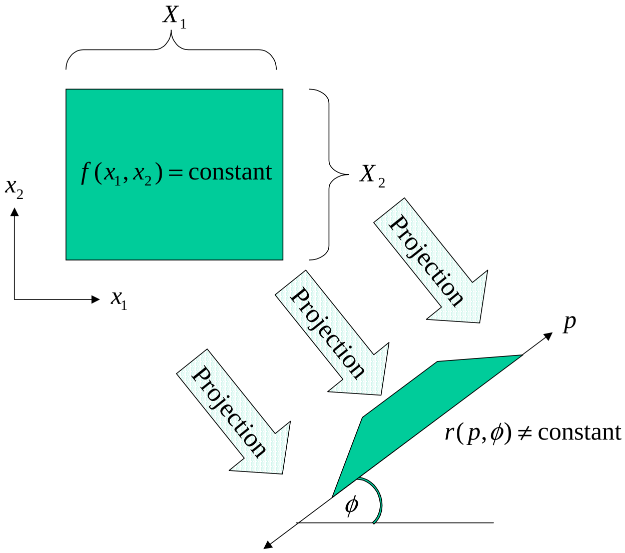
X_1
$f(x_1, x_2) = \text{constant}$
X_2
x_2
x_1
Projection
Projection
Projection
p
$r(p, \phi) \neq \text{constant}$
ϕ

Figure 2.13: Projection of a constant image

where

$$S^+(p,\phi) \quad = \quad min\left[p\cot\phi + \frac{X_1}{2\sin\phi}, -p\tan\phi + \frac{X_2}{2\cos\phi}\right] \tag{2.38}$$

$$S^-(p,\phi) \quad = \quad max\left[p\cot\phi - \frac{X_1}{2\sin\phi}, -p\tan\phi - \frac{X_2}{2\cos\phi}\right] \tag{2.39}$$

Here, the functions S^+, S^- come from the edges of the rectangular image region. See Figure 2.14. Thus, $r(p, \phi)$ is a piecewise linear function whose derivative will not be zero. Of course, projections at 0 and 90 degrees do not suffer from this anomaly. We propose to normalize the projections such that the projection of a constant image will produce a constant 1-D function. To accomplish this we use a normalized Radon transform of the form

$$\check{r}(p,\phi) \equiv \check{\mathcal{R}}_\theta\left[f(x_1,x_2)\right] = \frac{\int\int f(x_1,x_2)\delta\left(p - x_1\cos\phi - x_2\sin\phi\right)dx_1\,dx_2}{S(\phi)} \tag{2.40}$$

After computing the normalized Radon transform, we compute the derivatives of the projection at a specific angle θ by

$$\tilde{r}_p(p,\phi) = \check{r}(p,\phi) * g(p) \tag{2.41}$$

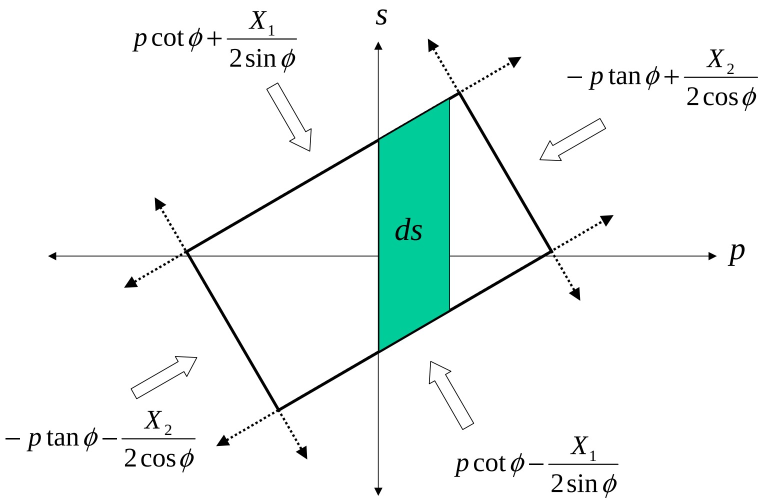
$p\cot\phi + \frac{X_1}{2\sin\phi}$
s
$-p\tan\phi + \frac{X_2}{2\cos\phi}$
ds
p
$-p\tan\phi - \frac{X_2}{2\cos\phi}$
$p\cot\phi - \frac{X_1}{2\sin\phi}$

Figure 2.14: Integration Region

where $g(p)$ represents the derivative convolution kernel. This will ensure that the proper spatial derivatives are calculated in the projection based motion estimators.

Chapter 3

Performance Analysis of Image Registration

In the last chapter, we detailed an efficient mechanism for drastically decreasing computational complexity while preserving and even improving performance. When evaluating the performance of such estimators, a natural question arises regarding the the significance of our improvement. To formalize the process of algorithm development, we must understand the fundamental limitations inherent to the problem of motion estimation. In this chapter, we study such performance limitations for the most basic form of motion estimation, namely, translation between a pair of frames. As we noted in Chapter 1, the translational model plays a significant role in a variety of imaging scenarios. This makes it a natural starting point when dealing with the complicated nonlinear estimation problem that is motion estimation. In addition, studying the two frame or image pair scenario not only offers insight into the more general problem of motion estimation, but also addresses a very practical field of motion estimation known as image registration.

The overall goal of this chapter is to quantify bounds on performance in estimating image translation between a pair of images. Such analysis lays the foundation for the bounds on multi-frame motion estimation performance bounds studied in Chapter 5. Because the problem of image registration is of such fundamental importance, many estimation algorithms have been developed over the years. In fact, there have been fairly comprehensive survey papers describing and comparing the performance of such algorithms, including [14], [44], and [45]. Unfortunately, the benchmarks comparing the performance of such algorithms tend to preclude the application of rigorous statistical analysis. These performance measures have ranged from geometric error criteria such as the mean angular error [14] shown in the last chapter, to visual inspection of the vector field for situations where ground truth is not available. While these measures have been very useful in advancing the methodology of motion estimation, they fail to evaluate estimator performance from a statistically interpretable perspective. Furthermore, the performance evaluation has relied on comparison between different algorithms, leaving open the important question of how close the algorithms come to achievable limits.

The problem of translational motion estimation is analogous to the classical problem of time delay estimation (TDE) as found in the signal processing literature [46]. For the TDE problem, performance is measured based on the mean square error (MSE) of a given estimator. In this chapter, we study the performance of image registration algorithms using this measure. By using MSE we can explore the fundamental performance bounds using the Cramér-Rao inequality. Surprisingly, while the Cramér-Rao inequality has been used widely in the field of time delay estimation in communication, Radar, and Sonar, except for a few isolated attempts [47], [48], it has not been utilized to understand the problem of image registration in general. In this chapter, we analyze the form of the Cramér-Rao inequality as it relates to the specific problem of registering translated images that have been sampled above the Nyquist rate. As a precursor to Chapter 5, we also introduce the extension to the case where the image is sampled below the Nyquist rate.

Developing such performance bounds provides a mechanism for critically comparing the performance of algorithms. We will show how a great deal of the heuristic knowledge used in motion estimation can be explained by examining this performance bound. Furthermore, understanding these fundamental limitations provides better understanding of the limitations inherent to the class of image processing problems that require image registration as a preprocessing step. In addition, analyzing the details of the bound offers insight into the very nature of the problem itself, thereby suggesting methods for improved algorithm design. In particularly, we will present the inherent performance tradeoff between bias and variance for several popular motion estimators.

This chapter is organized into three sections. In Section 3.1, we introduce the Cramér-Rao inequality. In Section 3.2, we derive the performance bounds in registering translated images, based on the Cramér-Rao inequality. We show how these bounds depend on image content by analyzing the Fisher Information matrix. We show the inherent problem of bias for the problem of image registration. In Section 3.3, we present experimental evidence of such bias for several popular estimation algorithms.

3.1 Introduction to the Cramér-Rao Bound

In this section, we introduce the Cramér-Rao lower bound (CRB) which we will use to quantify the fundamental MSE performance bounds on image registration. We will use this bound again in Chapter 5 to address a related estimation problem. Essentially, the CRB characterizes, from an information theoretic standpoint, the *difficulty* with which a set of parameters can be estimated by examining the given data model. In general, the CRB provides the lower bound on the mean square error (MSE) of *any* estimate $\hat{\mathbf{\Phi}}$ of an unknown parameter vector $\mathbf{\Phi}$ from a given set of measured data denoted $\mathbf{Z}$. Specifically, the Cramér-Rao bound on the error correlation matrix $E[(\hat{\mathbf{\Phi}}-\mathbf{\Phi})(\hat{\mathbf{\Phi}}-\mathbf{\Phi})^T]$ for any estimator is given by

$$MSE(\mathbf{\Phi}) \geq \frac{\partial E[\hat{\mathbf{\Phi}}]}{\partial \mathbf{\Phi}} \mathbf{J}^{-1}(\mathbf{\Phi}) \frac{\partial E[\hat{\mathbf{\Phi}}]}{\partial \mathbf{\Phi}}^T + (E[\hat{\mathbf{\Phi}}]-\mathbf{\Phi})(E[\hat{\mathbf{\Phi}}]-\mathbf{\Phi})^T \tag{3.1}$$

where the matrix $\mathbf{J}(\mathbf{\Phi})$ is referred to as the Fisher Information Matrix (FIM), and $E[\hat{\mathbf{\Phi}}]-\mathbf{\Phi}$ represents the bias of the estimator [49]. We refer to the error correlation matrix as $MSE(\mathbf{\Phi})$ since the diagonal terms of $E[(\hat{\mathbf{\Phi}}-\mathbf{\Phi})(\hat{\mathbf{\Phi}}-\mathbf{\Phi})^T]$ represent the MSE of the individual parameter components. The inequality indicates that the difference between the MSE (left side) and the

CRB (right side) will be a positive semidefinite matrix. From this formulation, we see that the mean square error bound is comprised of two terms corresponding to a variance term and a term which is the square or outer product of the of the bias associated with the estimator. Ideally, we could construct an estimator devoid of bias. Assuming such an estimator exists, the bound (3.1) simplifies to the more familiar

$$MSE(\mathbf{\Phi}) \quad \geq \quad \mathbf{J}^{-1}(\mathbf{\Phi}) \tag{3.2}$$

Thus, for any unbiased estimator, $J(\mathbf{\Phi})$ characterizes the minimum variance (and hence MSE) attainable.

The Fisher Information Matrix **J** for an unknown deterministic parameter is given by

$$\{\mathbf{J}\}_{i,j} \quad = \quad -E\left[\frac{\partial^2 l(\mathbf{\Phi}|\mathbf{Z})}{\partial \Phi_i \partial \Phi_j}\right]. \tag{3.3}$$

where $l(\mathbf{\Phi}|\mathbf{Z})$ is the log-likelihood of the measured data **Z** for a given value of the unknown parameter **Φ**. The log-likelihood function is defined as

$$l(\mathbf{\Phi}|\mathbf{Z}) = \ln\left(\mathrm{pdf}_{\mathbf{Z}}(\mathbf{Z}|\mathbf{\Phi})\right) \tag{3.4}$$

where $\mathrm{pdf}_{\mathbf{Z}}(\mathbf{Z}|\mathbf{\Phi})$ is the probability density function (pdf) of the measured data **Z** given the set of parameters **Φ**. Such a function gives the probability that the observed data was produced by a model with the particular set of parameters. If the unknown parameter vector is stochastic with a certain log-prior distribution $l(\mathbf{\Phi})$, the FIM is given by

$$\begin{aligned} \{\mathbf{J}\}_{i,j} \quad &= \quad -E\left[\frac{\partial^2 l(\mathbf{\Phi},\mathbf{Z})}{\partial \Phi_i \partial \Phi_j}\right] \\ &= \quad -E\left[\frac{\partial^2 l(\mathbf{\Phi}|\mathbf{Z})}{\partial \Phi_i \partial \Phi_j} + \frac{\partial^2 l(\mathbf{\Phi})}{\partial \Phi_i \partial \Phi_j}\right] \\ &= \quad \{\mathbf{J}_d\}_{i,j} + \{\mathbf{J}_p\}_{i,j} \end{aligned} \tag{3.5}$$

We use the subscripts d to denote information arising from the measured data and p to denote the information from the prior [49]. For many inverse problems including motion estimation, the data information matrix $\mathbf{J}_d$ can be very poorly conditioned or even rank deficient. In such situations, prior knowledge about the unknown parameter is key to solving the problem. Thus, prior information does not allow one to break the performance limits, but instead makes the fundamental limit more favorable. Such prior information will become essential in Chapter 5.

One important property of the CR bound is that if we are interested in estimating some function (possibly a vector valued function) $\chi(\mathbf{\Phi})$ of the unknown parameter vector, the CR bound for estimating the unknown vector in the new parameter space is given by

$$MSE(\chi(\hat{\mathbf{\Phi}})) \geq \nabla\chi(\mathbf{\Phi})\mathbf{J}^{-1}(\mathbf{\Phi})\nabla\chi(\mathbf{\Phi})^T \tag{3.6}$$

where $\nabla\chi(\mathbf{\Phi})$ denotes the gradient of the function $\chi(\mathbf{\Phi})$. We will exploit this property later in Chapter 5.

Often it is more convenient to evaluate estimation performance for vector valued parameters using a scalar measure of performance. We propose measuring estimator performance by

$$\overline{rmse}(\mathbf{\Phi}) = \sqrt{\frac{Tr(MSE(\mathbf{\Phi}))}{d}} \tag{3.7}$$

where d is the dimension of the unknown parameter vector $\mathbf{\Phi}$. Such a performance measure is useful when every element of the parameter vector of interest has the same units. The$\overline{rmse}(\mathbf{\Phi})$ has the interpretation of being the overall MSE averaged over the set of unknown parameters. The square root ensures that the performance measure is in the same units as the unknown parameters. Correspondingly, we may modify the CR inequality to bound this performance measure as well. We use the following notation to capture this bound. For the class of unbiased estimators, the bound becomes

$$T(\mathbf{\Phi}) = \sqrt{\frac{Tr(\mathbf{J}^{-1}(\mathbf{\Phi}))}{d}}. \tag{3.8}$$

For the class of biased estimators, we must use the complete CR bound whose corresponding scalar performance measure is given by

$$T(\mathbf{\Phi}) = \left[\frac{1}{d}Tr\left(\frac{\partial E[\hat{\mathbf{\Phi}}]}{\partial\mathbf{\Phi}}\mathbf{J}^{-1}(\mathbf{\Phi})\frac{\partial E[\hat{\mathbf{\Phi}}]}{\partial\mathbf{\Phi}}^T\right) + \frac{1}{d}(E[\hat{\mathbf{\Phi}}] - \mathbf{\Phi})^T(E[\hat{\mathbf{\Phi}}] - \mathbf{\Phi})\right]^{\frac{1}{2}}. \tag{3.9}$$

The CR bound for the overall performance measure is expressed as

$$\overline{rmse}(\mathbf{\Phi}) \geq T(\mathbf{\Phi}) \tag{3.10}$$

Such a performance bound has been justified and used in the past [50].

Finally, to address the utility of the CR bound in studying general estimation problems, we note that the overall usefulness of a performance limit depends on its ability not only to limit, but predict actual estimator performance. For example, we might trivially bound MSE performance as $MSE(\mathbf{\Phi}) \geq 0$. While such a bound is provably correct, it offers no useful information about the estimation problem. The CR bound, however, can be shown theoretically to be asymptotically attainable by the class of Maximum Likelihood (ML) estimators. While there is no guarantee that such estimators are realizable, it does offer hope for predicting performance for a wide class of estimators.

3.2 Performance Limits in Image Registration

In this section we derive the Fisher Information Matrix for the problem of image registration. Analysis of the Fisher Information Matrix for image registration reveals interesting structure associated with the nonlinear image registration problem.

3.2.1 Fisher Information for Image Registration

The Fisher Information matrix provides a measure of the influence an unknown parameter vector has in producing observable data. In our case, the unknown vector is the translation vector $\mathbf{v}_0 = [v_{0_1} \ v_{0_2}]^T$. The FIM is derived by looking at the expected concavity of the likelihood function. Intuitively, a likelihood maximizing estimator should have an easier time finding the maximum of a sharply peaked likelihood function than a rather flat one.

We assume in this chapter that we are given only a pair of images with which to estimator $\mathbf{v}_0$. We relate this data model to our original motion model by

$$
\begin{aligned}
z_0(n_1, n_2) &= z(n_1, n_2, 0) \\
z_1(n_1, n_2) &= z(n_1, n_2, 1).
\end{aligned}
$$

We model the noise as being additive gaussian noise with zero mean and variance σ^2.

The conditional log-likelihood function for our data is given by

$$
\begin{aligned}
l(z|\mathbf{v}_0) = \frac{-1}{2\sigma^2} \sum_{n_1,n_2} & [z_0(n_1, n_2) - f(n_1, n_2)]^2 + \\
& [z_1(n_1, n_2) - f(n_1 - v_{0_1}, n_2 - v_{0_2})]^2 + const. \qquad (3.11)
\end{aligned}
$$

The Fisher Information matrix measures the sharpness or curvature of likelihood peak as defined by equation (3.3). In deriving the FIM, we first compute the partial derivatives with respect to the log-likelihood function:

$$
\begin{aligned}
\frac{\partial^2 l(z|\mathbf{v}_0)}{\partial v_i^2} &= \frac{\partial}{\partial v_{0_i}} \left[\frac{1}{\sigma^2} \sum_{n_1,n_2} \left(z_1 - \tilde{f} \right) \frac{\partial \tilde{f}}{\partial v_{0_i}} \right] \\
&= \frac{1}{\sigma^2} \sum_{n_1,n_2} \left[\left(z_1 - \tilde{f} \right) \frac{\partial^2 \tilde{f}}{\partial v_{0_i}^2} - \left(\frac{\partial \tilde{f}}{\partial v_{0_i}} \right)^2 \right] \qquad (3.12)
\end{aligned}
$$

To simplify the notation, we refer to the transformed image $f(n_1 - v_{0_1}, n_2 - v_{0_2})$ as $\tilde{f}$. Since only the term z_1 is random, the negative expectation of (3.12) for each term becomes

$$
\begin{aligned}
-E\left[\frac{\partial^2 \log \mathbf{P}(z; \mathbf{v}_0)}{\partial v_{0_1}^2} \right] &= \frac{1}{\sigma^2} \left(\frac{\partial \tilde{f}}{\partial v_{0_1}} \right)^2 \\
-E\left[\frac{\partial^2 \log \mathbf{P}(z; \mathbf{v}_0)}{\partial v_{0_2}^2} \right] &= \frac{1}{\sigma^2} \left(\frac{\partial \tilde{f}}{\partial v_{0_2}} \right)^2 \\
-E\left[\frac{\partial^2 \log \mathbf{P}(z; \mathbf{v}_0)}{\partial v_{0_2} \partial v_{0_1}} \right] &= \frac{1}{\sigma^2} \left(\frac{\partial \tilde{f}}{\partial v_{0_2}} \right) \left(\frac{\partial \tilde{f}}{\partial v_{0_1}} \right).
\end{aligned}
$$

Finally, the chain rule implies

$$
\begin{aligned}
\frac{\partial \tilde{f}}{\partial v_{0_1}} &= \frac{\partial \tilde{f}}{\partial x_1} = f_{x_1}(n_1 - v_{0_1}, n_2 - v_{0_2}) \\
\frac{\partial \tilde{f}}{\partial v_{0_2}} &= \frac{\partial \tilde{f}}{\partial x_2} = f_{x_2}(n_1 - v_{0_1}, n_2 - v_{0_2}).
\end{aligned}
$$

Hence, we get the Fisher Information matrix

$$\mathbf{J}(\mathbf{v}_0) = \frac{1}{\sigma^2}\begin{bmatrix} J_{v_1,v_1} & J_{v_1,v_2} \\ J_{v_1,v_2} & J_{v_2,v_2} \end{bmatrix} \tag{3.13}$$

where

$$\begin{aligned} J_{v_1,v_1} &= \sum_{n_1,n_2} f_{x_1}^2(n_1 - v_{0_1}, n_2 - v_{0_2}) \\ J_{v_1,v_2} &= \sum_{n_1,n_2} f_{x_1}(n_1 - v_{0_1}, n_2 - v_{0_2}) f_{x_2}(n_1 - v_{0_1}, n_2 - v_{0_2}) \\ J_{v_2,v_2} &= \sum_{n_1,n_2} f_{x_2}^2(n_1 - v_{0_1}, n_2 - v_{0_2}) \end{aligned}$$

The subscripts indicate the partial derivative in the x_1, x_2 direction.

A comment is in order regarding these partial derivatives. The Fisher Information matrix, and hence the performance bound, depend on the partial derivatives of the shifted version of the continuous image $f(x_1, x_2)$ evaluated at the sample locations n_1, n_2. While this is simple to present theoretically, in practice, the partial derivatives of the image function are not available. In fact, only samples of the image function are available, which presents a practical challenge when trying to compute the Fisher Information matrix. There are a few approximations that can be made in order to calculate the FIM depending on the information available prior to estimation. For instance, if a relatively noise-free image is available, preferably of higher resolution than the images being registered, then the partial derivatives may be approximated using derivative filters. For situations where the scene being observed is known prior to estimation, such as in industrial applications, a continuous image function can be constructed to represent the scene and differentiated analytically. Finally, if only the discrete images are available, then such an image function can be approximated directly from the samples. One such method assumes that the image can be expressed as a Fourier series of the form

$$f(x_1, x_2) = \sum_{n_1}^{N}\sum_{n_2}^{N} F\left(\frac{2\pi n_1}{N}, \frac{2\pi n_2}{N}\right) e^{j2\pi\left(\frac{x_1 n_1}{N} + \frac{x_2 n_2}{N}\right)} \tag{3.14}$$

where $F\left(\frac{2\pi n_1}{N}, \frac{2\pi n_2}{N}\right)$ are the coefficients of the discrete Fourier transform (DFT) of the image.

We use this last assumption throughout this chapter in our experiments. By construction, this guarantees that the image is sampled above the Nyquist rate.

3.2.2 Analysis of the FIM for Image Registration

To gain further insight, we now consider the FIM in the Fourier domain. To do so, we first must make certain general assumptions about our underlying image function $f(x_1, x_2)$. In particular, we assume that the image function is bandlimited and is sampled at a rate greater than Nyquist. Then, the discrete time Fourier transform (DTFT) of the samples of the derivative function $f_{x_1}(n_1 - v_{0_1}, n_2 - v_{0_2})$ can be written as $e^{j(v_{0_1}\theta_1 + v_{0_2}\theta_2)} j\theta_1 F(\theta_1, \theta_2)$ and similarly for the x_2 partial derivative. With such an image model, we then can write the terms of the FIM using Parseval's relation:

$$\begin{aligned}
J_{v_1,v_1} &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |F(\theta_1,\theta_2)|^2 \theta_1^2 d\theta_1 d\theta_2 \\
J_{v_1,v_2} &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |F(\theta_1,\theta_2)|^2 \theta_1 \theta_2 d\theta_1 d\theta_2 \\
J_{v_2,v_2} &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |F(\theta_1,\theta_2)|^2 \theta_2^2 d\theta_1 d\theta_2.
\end{aligned}$$

Examining the FIM using this formulation, we see that it does not depend on the unknown translation vector $\mathbf{v}_0$ and depends only on the image content. This observation depends on our assumption that the image is periodic outside the field of view. This independence of the FIM on $\mathbf{v}_0$ no longer holds when the image is sampled below the Nyquist rate. When the images to be registered are aliased, as we shall show in Chapter 5, the FIM depends on the unknown motion $\mathbf{v}_0$.

It is interesting to note that one can explain the well-known *aperture problem* [14] by examining the FIM. This problem arises when the spectral content of the image is highly localized. An example of this occurs when all of the spectral energy is contained along a slice passing through the origin of the spectrum at an angle ψ_0. Equivalently, in the spatial domain, the texture of the image is one-dimensional in nature. Figure 3.1 shows an example of such images in both the spatial and frequency domain.

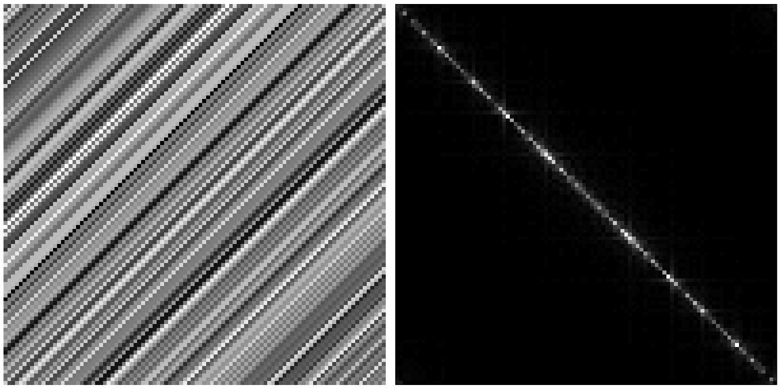

Figure 3.1: Example of the aperture effect in spatial (left) and frequency (right) domain

In polar coordinates, such a spectrum looks like

$$F(\psi,\rho) = \begin{cases} F(\psi_0 \ ,\rho), & \psi = \psi_0 \\ 0\ , & else \end{cases} \tag{3.15}$$

The terms of the corresponding FIM in polar coordinates are,

$$\begin{aligned} J_{v_1,v_1} &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi} |F(\psi_0,\rho)|^2\rho^3\cos^2\psi_0 d\rho \\ J_{v_1,v_2} &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi} |F(\psi_0,\rho)|^2\rho^3\cos\psi_0\sin\psi_0 d\rho \\ J_{v_2,v_2} &= \frac{1}{4\pi^2}\int_{-\pi}^{\pi} |F(\psi_0,\rho)|^2\rho^3\sin^2\psi_0 d\rho \end{aligned}$$

Since the determinant of the FIM is

$$\begin{aligned} \det \mathbf{J}(\mathbf{v}_0) &= J_{v_1,v_1}J_{v_2,v_2} - J_{v_1,v_2}^2 \\ &= (\cos^2\psi_0\sin^2\psi_0 - \cos^2\psi_0\sin^2\psi_0)K = 0, \end{aligned}$$

(where K is a constant), $\mathbf{J}(\mathbf{v}_0)$ is therefor not invertible, and any unbiased estimator will have infinite variance. Essentially, there is not enough information with which to register the pair of images.

Next, we further observe that the information contained in a pair of images depends only on the gradients or the texture of the image. The relationship between estimator performance and image content has been noted in previous works and used to select features to

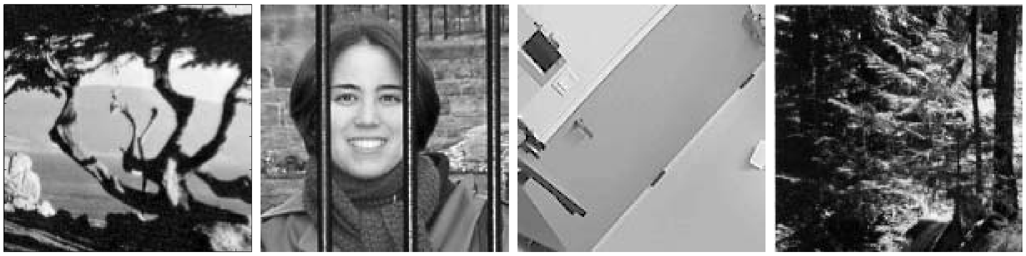

Figure 3.2: Experimental images (Tree, Face, Office, Forest)

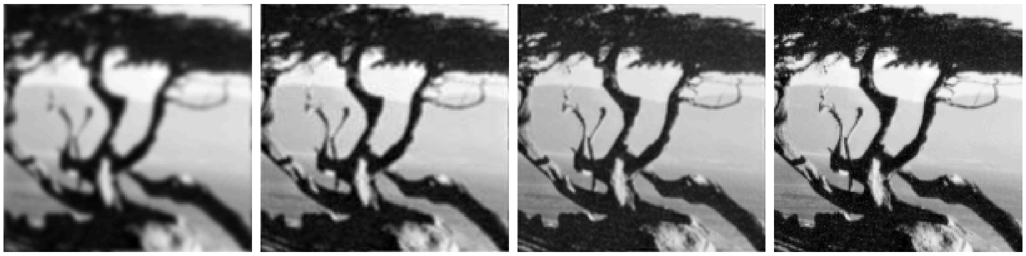

Figure 3.3: Tree image filtered filtered by low-pass filters with cutoff frequencies of 25, 50, 75, 100 $\%$.

register [51]. This previous work, however, provided only the heuristic suggestion that features with high frequency content are better for tracking by looking at one specific estimator. Here, we suggest the performance bound $T(\mathbf{v}_0)$ as a scalar predictor of performance as it relates to image content. In general, as $T(\mathbf{v}_0)$ decreases, improved estimator performance is expected. Figure 3.4 shows $T(\mathbf{v}_0)$ vs image bandwidth for the images shown in Figure 3.2.

The image spectral bandwidth was controlled by filtering the images with a low-pass filter whose radial cutoff frequency θ_c was constructed to be a percentage of the full image bandwidth. All of the images were normalized, in that they were cropped to the same size and scaled to have the same intensity range. As seen in Figure 3.4, $T(\mathbf{v}_0)$ decreases as the image bandwidth increases. This corroborates the general intuition that highly textured images are easier to register. For the purpose of intuition, Figure 3.3 shows an image with different cutoff frequencies. Furthermore, we see from Figure 3.4 that while the performance may continue to improve with greater frequency content, the improvement tapers off as the bandwidth increases beyond about a quarter of the full bandwidth. This observation might be explained by the$\frac{1}{\theta_c} =$

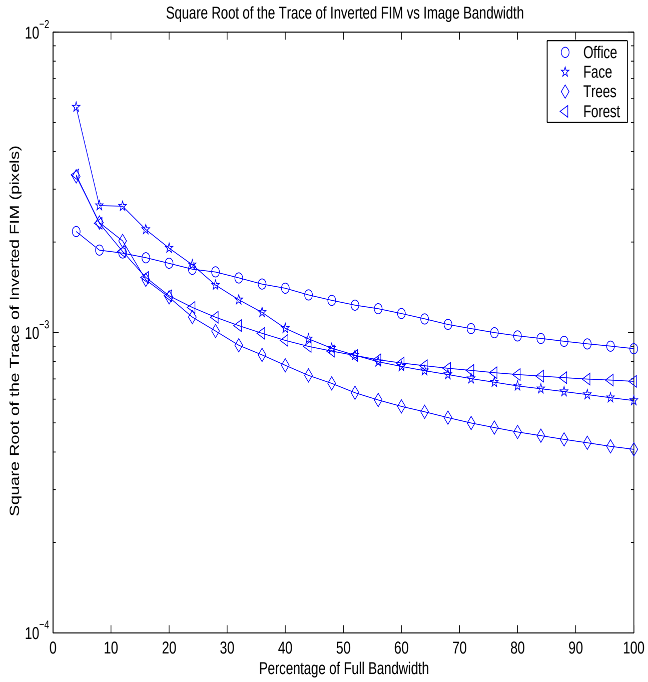

Figure 3.4: Trace of $\mathbf{J}^{-1}$ vs image bandwidth

$\frac{1}{\sqrt{\theta_1^2+\theta_2^2}}$ spectral amplitude decay commonly found in natural images [52]. This suggests that $T(\mathbf{v}_0)$ could be approximated by a term such as $\frac{1}{\theta_c}$ where θ_c is the radial cutoff frequency (or bandwidth of the image). Figure 3.4 exhibits a $\log\frac{1}{\theta_c}$ type behavior. These results also suggest that the inherent bandwidth limitations induced by the imaging system affect the fundamental performance limits for image registration. Since the spectral bandwidth of the image predicts the ability to register the image, the inherently bandlimited nature of imaging systems eventually dominates the achievable performance limits.

Another interesting way to explore the registration performance limits as a function of image content is by examining the bounds along particular directions. Instead of estimating both the v_{0_1} and v_{0_2} components of translation, we consider the linear combination $v_\phi = v_{0_1}\cos\phi + v_{0_2}\sin\phi = \mathbf{n}_\phi^T\mathbf{v}_0$ of the unknown parameters. The CRB inequality (3.2) can be extended to bound the performance in estimating a linear combination of the unknown parameters using (3.6). In particular, we have $Var(\mathbf{n}_\phi^T\mathbf{v}_0) \geq \mathbf{n}_\phi^T J^{-1}(\mathbf{v}_0)\mathbf{n}_\phi$. From this inequality,

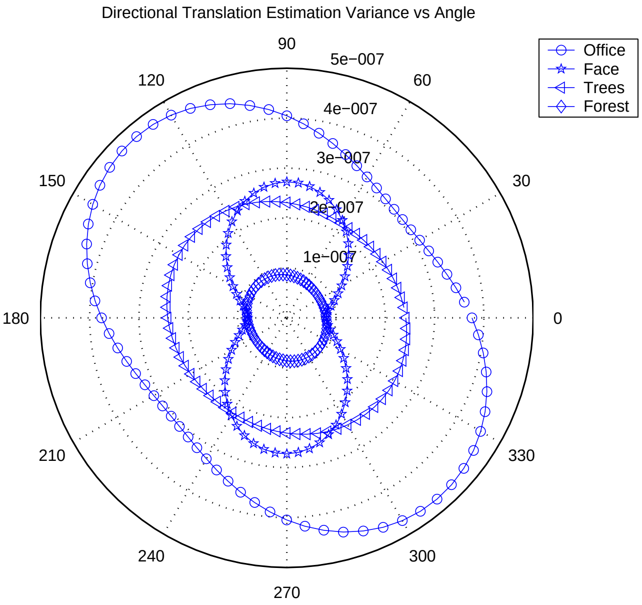

Figure 3.5: Angular estimation information as a function of image content

it becomes apparent that, for a particular image, certain angles have better inherent performance - these optimal angles depending on the eigenvectors of the matrix $\mathbf{J}^{-1}(\mathbf{v}_0)$. Figure 3.5 shows the variance bound on the estimation of the directional components of translation as a function of angular direction for the four example images in Figure 3.2. The face image and, to a lesser extent, the office image, have specific directions in which estimates are most reliable. Specifically, the vertical bars in the face image provide large amounts of spectral energy in the x_1 direction. This spectral signature correspondingly suggests small estimator variance in this angular direction. Similarly, the office image is rotated about 45 degrees, so the dominant derivative energy is located around 45 degrees.

3.3 Bias in Image Registration Algorithms

In this section, we show that many of the current algorithms used to solve the inverse problem of image registration are inherently biased. This implies that the bound given by (3.2) is overly optimistic and the complete bound (3.1) must be used to accurately predict estimator performance. Finally, it shows that many of the currently popular estimators would benefit from further study.

To understand the inherent bias associated with any translational motion estimator, we look at the class of maximum likelihood (ML) estimators. Many image registration algorithms can be shown to produce approximate solutions to the maximum likelihood equation. To find the ML solution, we again look at the log likelihood function for the shift parameters

$$l(z|\mathbf{v}_0) = \frac{-1}{2\sigma^2} \sum_{n_1,n_2} [z_0(n_1,n_2) - f(n_1,n_2)]^2 + \\ [z_1(n_1,n_2) - f(n_1 - v_{0_1}, n_2 - v_{0_2})]^2 + const.$$

Since only the second term depends on the unknown parameters, the maximization problem can be expressed as a minimization of the objective function

$$\mathcal{C}_{LS}(\mathbf{v}_0) \quad = \quad \sum_{n_1,n_2} [z_1(n_1,n_2) - f(n_1 - v_{0_1}, n_2 - v_{0_2})]^2 . \tag{3.16}$$

This is the general nonlinear least squares objective function used in defining the ML solution. By expanding the quadratic in (3.16) we get

$$\sum_{n_1,n_2} \left[z_1^2(n_1,n_2) - 2z_1(n_1,n_2) f(n_1 - v_{0_1}, n_2 - v_{0_2}) + f^2(n_1 - v_{0_1}, n_2 - v_{0_2})\right] . \tag{3.17}$$

Ignoring the first term since it does not depend on the parameter $\mathbf{v}_0$, and negating the entire function we can rewrite the objective function as

$$\textstyle\sum_{n_1,n_2} 2z_1(n_1,n_2) f(n_1 - v_{0_1}, n_2 - v_{0_2}) - \sum_{n_1,n_2} f^2(n_1 - v_{0_1}, n_2 - v_{0_2}). \tag{3.18}$$

By normalizing the entire cost function with respect to the energy in the image, (the second term of (3.18)), we obtain the direct correlator objective function

$$\mathcal{C}_{DC}(\mathbf{v}_0) \quad = \quad \frac{\sum_{n_1,n_2} z_1(n_1,n_2) f(n_1 - v_{0_1}, n_2 - v_{0_2})}{\sum_{n_1,n_2} f^2(n_1 - v_{0_1}, n_2 - v_{0_2})}. \tag{3.19}$$

In general, minimizing/maximizing these two objective functions with respect to the unknown parameter $\mathbf{v}_0$ provides the ML solution.

As previously noted, however, the function $f(n_1 - v_{0_1}, n_2 - v_{0_2})$ is typically unknown. An approximate ML solution is found using an estimate of the unknown function, most commonly given by $z_0(n_1 - v_{0_1}, n_2 - v_{0_2})$. In essence, the measured reference image $z_0(n_1, n_2)$ becomes an estimate of the unknown image $f(n_1, n_2)$. It is easy to see that at very high SNR, this estimate should be very close to $f(n_1 - v_{0_1}, n_2 - v_{0_2})$. As we shall see in Chapter 5, however, when the images become aliased, a single measured image is an insufficient estimate of the function $f(n_1, n_2)$. Even in such high SNR (low noise) situations, however, the objective functions (3.16) and (3.19) can be evaluated only for integer values of v_{0_1} and v_{0_2}, constraining the estimates to that of integer multiples of pixel motion. While some progress has been made to address this issue [46], [53], [45], the proposed algorithms often are based on overly simplified approximations that are known to produce biased estimates [54].

For many applications in image processing, accurate subpixel image registration is needed. To register images to subpixel accuracy, the image function $f(x_1, x_2)$ effectively must be reconstructed from the noisy samples of $z_0(n_1, n_2)$. In general, this reconstruction is an ill-posed problem. All estimators contain inherent prior assumptions about the space of continuous images under observation. These priors act to regularize the problem, allowing solutions to be found. But, when the real underlying functions do not match the model assumptions, the estimators inevitably produce biased estimates. There is only a small class of images for which the problem is not ill-posed. The exception occurs when the underlying continuous image is *constructed* through the assumed forward model such as (3.14). Unfortunately, this requirement is not likely to satisfied in general image processing scenarios, implying that estimation algorithms may often be inherently biased. As we shall show, however, even under ideal conditions, many of the current estimation algorithms contain bias.

To verify the presence of this bias in existing algorithms, we conduct a Monte-Carlo simulation computing actual estimator performance for a collection of image registration algorithms. The estimators used in the experiment are the following.

1. **Approximate Minimum Average Square Difference (ASD) (2-D version of [46]) -**

Samples of the average square difference function,

$$ASD(v_{0_1}, v_{0_2}) = \frac{1}{MN} \sum_{n_1,n_2} (z_0(n_1 - v_{0_1}, n_2 - v_{0_2}) - z_1(n_1, n_2))^2 \tag{3.20}$$

(an approximation to (3.16)) are computed for pixel shift values of v_{0_1} and v_{0_2} in some range. Then, the subpixel shift is computed by finding the minimum of a quadratic fit about the minimum of the cost function given for integer pixel shifts.

2. **Approximate Maximum Direct Correlator (DC) [45] -** A sample correlation estimate is used to approximate (3.19). Essentially, the denominator of (3.19) is assumed to be approximately constant, independent of the underlying image shift $\mathbf{v}_0$. Thus, the simplified sample correlation estimate

$$Cor(v_{0_1}, v_{0_2}) = \frac{1}{MN} \sum_{n_1,n_2} z_0(n_1 - v_{0_1}, n_2 - v_{0_2}) z_1(n_1, n_2) \tag{3.21}$$

is computed for integer pixel shifts. Then, the subpixel shift is estimated as the maximum of a quadratic fit about the maximum of the sample correlation function.

3. **Gradient-Based Method (GB)-** This method was introduced in Chapter 2.

4. **Multiscale (Pyramid) Gradient-Based Method (Pyr)-** For this method, we utilized a multiscale pyramid with 3 levels. At each iteration the 2-D gradient-based method was applied to estimate translation.

5. **Projection Gradient-Based Method (Proj-GB)-** This is the projection-based method introduced in Chapter 2. For our experiments, we used only a pair of image projections at 0 and 90 degrees (x_1, x_2 axes).

6. **Projection Multiscale Gradient-Based Method-** Again, this is the multiscale method introduced in Chapter 2, using 3 levels in a multiscale pyramid. A pair of image projections was used at 0 and 90 degrees.

7. **Relative Phase (Phase) [43]-** Using the shift property of the Fourier transform it is noted that

$$\frac{F_0 F_1^*}{|F_1|^2} = \frac{F_0}{F_1} = e^{j2\pi(\theta_1 v_{0_1} + \theta_2 v_{0_2})}. \tag{3.22}$$

where F_0 and F_1 denote the Fourier transform of the image function $f(x_1, x_2)$ and a shifted version $f(x_1 - v_{0_1}, x_2 - v_{0_2})$ respectively. The vector $\mathbf{v}_0$ is estimated by finding the solution to the set of linear equations of the phase function

$$\angle\frac{Z_0}{Z_1} = j2\pi(\theta_1 v_{0_1} + \theta_2 v_{0_2}) \tag{3.23}$$

where $Z_{1,2}$ represents the DFT of the input images $z_{1,2}$ and $\angle$ indicates the measured phase angle. We used the implementation of [43] wherein the solution is found using weighted least squares.

To generate a pair of images for the experiment, we use the discrete Fourier transforms (DFT) approach following the method of [55]. This effectively generates an image pair, assuming the continuous model is given by (3.14). Such a model is necessary given that we want to focus on the problem of estimating sub-pixel shifts. Furthermore, such a motion model is entirely reasonable for a large image, where the modelling error associated with the assumption that the image is periodic is negligible. We used the Tree image from [14], which is of dimension 150×150, in the experiment. As the image region shrinks, the assumption that the image region is periodic outside the region of observation is less likely to represent the image data accurately. In this sense, our experimental setup examines a scenario where highly accurate estimation is expected.

To synthesize the effects of noise in the imaging system, we add white Gaussian noise to the image pair prior to estimation and the entire process was repeated 500 times at each SNR value. We explore SNR values from 0 dB (very noisy) to 70 dB (effectively noiseless). To capture a single representation of error, we compute $\overline{rmse}(\mathbf{v}_0)$. Figure 3.6 shows this measure of actual estimator performance as a function of of SNR for the estimators mentioned above. The dashed line indicates the predicted performance using $T(\mathbf{v}_0)$ for the class of unbiased estima-

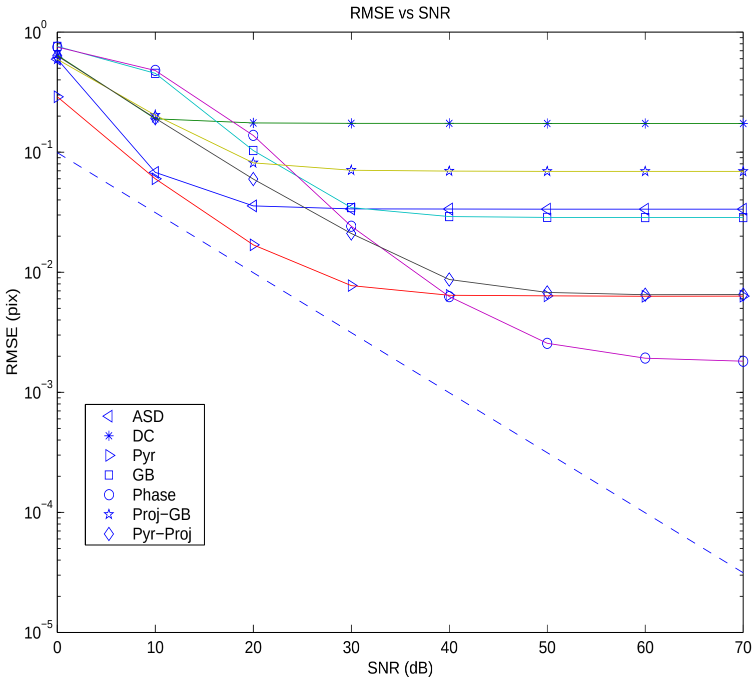

Figure 3.6: Magnitude error performance vs SNR $\mathbf{v}_0 = [.5 \;\; .5]^T$

tors. While this bound suggests continued improvement as the noise decreases, above certain SNR values, the performance of each estimator levels out. This flattening of the performance curves is indicative of the bias present in each of the estimators.

Immediately, we observe a certain bias-variance tradeoff between various algorithms. For instance, the multiscale 2-D gradient-based algorithm appears to offer superior performance for the low SNR situations. At higher SNR, the phase-based method offers better performance. However, at lower SNR, the phase-based approach is one of the worst estimators in the group, suggesting a high sensitivity to noise. The multiscale gradient-based approach is less sensitive to noise, but ultimately suffers from worse estimator bias.

While we can see the effect of this bias experimentally, the actual bias function for a given estimator typically is very difficult to express. The overall bias is often a combination of both the deterministic modelling error and the statistical bias of the estimator. If the estimator is an ML estimator, the estimates theoretically should be asymptotically unbiased, leaving only the bias stemming from modelling error. This appears to be the dominant bias for high SNR as seen in Figure 3.6 where the bias is independent of the noise in the images. This modelling error has been addressed only infrequently in the image registration literature. In [56], the approximate direct correlation method (DC) produces biased estimates resulting from the quadratic approximation about the peak of the correlation function. Basically, the DC method using the quadratic approximation about the mean of the sample correlation function makes implicit assumptions about the underlying continuous function. In [56], and similarly in [54], the resulting bias is derived for situations where the likelihood function is not quadratic about its maximum as typically assumed. The gradient-based estimators have been studied in the context of bias as well [57], [58], [59], [55]. Nevertheless, an accurate functional expression describing the estimator bias is not available. In the next chapter, we describe these earlier attempts at understanding gradient-based estimator bias, and we derive and verify a new functional form of bias inherent to the class of gradient-based estimators.

3.4 Conclusion

In this chapter we derive the fundamental performance limits for translation estimation using the Cramér-Rao bound. In doing so, we have defended the idea that MSE should be used as a standard performance measure to prevent unfair comparisons between algorithms and to motivate statistically accurate analysis. We have shown that studying this performance bound, as it relates to image registration, provides much insight into the inherent tradeoffs between estimator variance and bias. We presented analysis as well as experimental evidence suggesting that a large class of motion estimators are in fact biased.

The analysis and experimentation presented in this chapter lay the foundation for rigorous statistical analysis of the motion estimation problem. The work opens several areas of further research. For instance, we focused on the estimation of translational motion. One could extend the analysis to more complex parametric motion models such as affine and bilinear motion. We hope that this type of analysis would offer guidance to the practitioner choosing between complex motion models for large image regions, or simple translational models for smaller or more local motion estimation.

Chapter 4

Gradient-Based Translation Estimation: A Case Study

In the last chapter, we compared the performance of several estimators with the fundamental performance limits. We observed that for a large class of estimators, bias dominated the MSE at higher SNR. While estimator bias is often difficult to express, in this chapter, we derive such bias expressions for the popular gradient-based estimator. While the bias for this class of estimators has been addressed in previous works [57], [58], [59], [55], [60] these works make overly simplified generalizations about the bias. In this chapter, we present and analyze more precise expressions for the estimator bias for high SNR situations. We will show that this bias limits overall estimation for typical imaging systems. Finally, we will use this bias function to propose a rule-of-thumb limit (based on our analytical results) for image registration accuracy using gradient-based estimators.

In addition, we show that having an expression of estimator bias allows the practitioner to optimize estimator performance. In particular, we show how we may improve gradient-based estimator performance through a careful design of the gradient filters.

4.1 Gradient-Based Estimator Bias

In this section we derive expressions for the bias associated with gradient-based estimators. To maintain clarity during the derivation, we focus on the 1-D analogue of gradient-based estimation. Since much of the derivation for the 1-D case is similar to the projection-based algorithm described in Chapter 2, we summarize 1-D estimator for a pair of signals as follows.

For the 1-D case, we suppose that the measured data is of the form

$$z_0(n) = f(n) + \epsilon_2(n) \tag{4.1}$$

$$z_1(n) = f(n + v_0) + \epsilon_1(n). \tag{4.2}$$

In the derivation of the gradient-based estimator, we must reformulate the data as $z(n) = z_0(n) - z_1(n) = f(n + v_0) - f(n) + \epsilon(n)$ where ϵ is a Gaussian white noise process with variance σ^2.

Gradient-based methods solve this equation for v_0 by linearizing the function $f(n + v_0)$ about a point $v_0 = 0$ in a Taylor series. This expansion looks like

$$f(n + v_0) - f(n) = v_0 f'(n) + R(n, v_0) \tag{4.3}$$

where R is the remainder term in the Taylor expansion. This remainder has the form $R(n, v_0) = \sum_{r=2}^{\infty} \frac{v_0^r}{r!} f^{(r)}(n)$. Thus, the new data model becomes $z(n) = v_0 f'(n) + R(n, v_0) + \epsilon(n)$. When the remainder term R is ignored, the linearized model of the data becomes $z(n) = f'(n)v_0 + \epsilon(n)$. Using the derivative values, we obtain the linear estimator for the velocity v_0 using least squares,

$$\hat{v}_0 = \frac{\sum f'(n)z(n)}{\sum (f'(n))^2}, \tag{4.4}$$

where the sum is taken to be over some region which may be the entire image. This type of estimator commonly is referred to as the optical flow, gradient-based, or differential estimation method [42], [2]. This estimator derivation assumes that in addition to the samples of $f(n)$, we also have samples of the derivative of the function $f'(n)$. Later, we show how this assumption is relaxed.

It is interesting to note that the variance of the gradient-based estimator is $var(\hat{v_0}) = \frac{\sigma^2}{\sum(f'(n))^2}$ if, in fact, the remainder term is zero. The variance is almost exactly the same as the CR bound introduced in Chapter3 for unbiased estimators, which is $\frac{\sigma^2}{\sum(f'(n+v_0))^2}$. This relationship implies that the gradient-based estimator would be a maximum likelihood estimator for the case when the remainder term is, in fact, zero.

4.1.1 Bias from Series Truncation

One source of systematic error or bias in the gradient-based estimation method comes from the remainder term $R(n, v_0)$ in (4.3), originally ignored to construct a linear estimator.

When we include the remainder term in the estimator, we obtain as the expected value of the estimator (4.4)

$$E[\hat{v}_0] = v_0 + \frac{\sum f'(n)R(n, v_0)}{\sum(f'(n))^2}. \tag{4.5}$$

So, unless the second term is zero, the higher order terms introduce a systematic bias into the estimator.

This is more informative in the frequency domain. First, we define the Fourier transform of the original function $f(x)$ as $F(\omega)$. Here, we assume that the image signal is bandlimited and has a cutoff spatial frequency of ω_c. Thus, for the signal to be sampled above the Nyquist rate, the sampling rate must satisfy $\frac{2\pi}{T_x} \geq 2\omega_c$, where ω_c is the cutoff frequency for the bandlimited signal. In other words, $F(\omega) = 0, \forall\omega \geq \omega_c$. Under the assumption that the function is sampled above the Nyquist rate, the DTFT of the derivative sequence $f'(n)$ can be represented as $j\theta F(\theta)$, where $\theta = \omega T_x$. By Parseval's relation, we can rewrite the estimator (4.4) as

$$\hat{v}_0 \quad = \quad \frac{\int_{-\pi}^{\pi} j\theta F(\theta)Z^*(\theta)d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta}. \tag{4.6}$$

As a side note, we can also arrive at the same estimator form by modelling the data itself directly in the frequency domain, as follows; the shifted sequence $f(n + v_0)$ has a DTFT of $F(\theta)e^{jv_0\theta}$ and the DTFT of the data model becomes

$$Z(\theta) \quad = \quad F(\theta)\left[e^{jv_0\theta} - 1\right] + \xi(\theta). \tag{4.7}$$

If we again expand the exponential in a Taylor series $e^{jv_0\theta} = 1 + jv_0\theta - \frac{(v_0\theta)^2}{2} + ...$ and truncate after the linear term, we obtain the linear relationship $Z(\theta) = [F(\theta)j\theta]\, v_0 + \xi(\theta)$ from which we obtain the linear estimator as (4.6).

Returning to the case where the complete data model is used, we see that the expected value of the estimate is

$$\begin{aligned} E[\hat{v}_0] &= \frac{\int_{-\pi}^{\pi} |F(\theta)|^2 j\theta(e^{-jv_0\theta} - 1)d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta} \\ &= \frac{\int_{-\pi}^{\pi} |F(\theta)|^2\theta\sin(v_0\theta)d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta} + j\frac{\int_{-\pi}^{\pi} |F(\theta)|^2\theta(\cos(v_0\theta) - 1)d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta} \\ &= \frac{\int_{-\pi}^{\pi} |F(\theta)|^2\theta\sin(v_0\theta)d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta} \end{aligned} \tag{4.8}$$

where in the last equality we note that since $Im[j\theta(e^{-jv_0\theta} - 1)] = \theta(\cos(v_0\theta) - 1)$ is an odd function, it integrates to zero. Using the expected value of the estimate, we obtain the bias function as follows

$$b(v_0) \quad = \quad E[\hat{v}_0] - v_0 = \frac{\int_{-\pi}^{\pi} |F(\theta)|^2\left(\theta\sin(v_0\theta) - v_0\theta^2\right)d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta}. \tag{4.9}$$

To verify this bias function experimentally, we measure the bias in estimating translation for a randomly constructed function such that the actual derivative values were available to the estimator. The actual function $f(n)$ used in the experiment is plotted in the left graph of Figure 4.1. The magnitude spectrum for the function used was $|F(\theta)| = \frac{1}{\theta}$ modelled after the spectrum of natural images. The phase angle of the Fourier spectrum was drawn from a uniform distribution in the range $[0, 2\pi]$. To measure the bias which is purely deterministic, no noise was added to the data prior to estimation. Figure 4.1 shows a plot of the experimental estimator bias as it depends on translation v_0. The plot shows three different curves which indicate the bias for the full bandwidth function $f(n)$ as well as two filtered versions of $f(n)$ wherein the functions

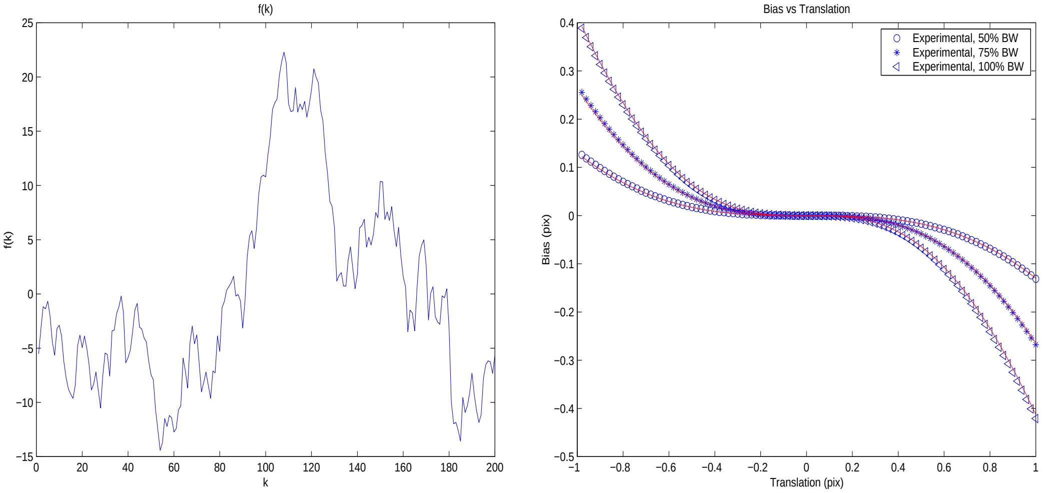

Figure 4.1: Plot of $f(n)$ (left) and estimator bias (right), continuous is predicted bias

were bandlimited to 50% and 75% of full signal bandwidth. The continuous curves represent the predicted performance using (4.9).

The bias function appears to follow the bias expression almost exactly. Furthermore, Figure 4.1 indicates that as the bandwidth of $f(n)$ increases, the bias becomes more severe. Here we immediately see a tradeoff with the the Fisher Information which suggests that increased bandwidth will improve estimator variance. We will examine this notion more closely later in Section 4.2.

We note that functional expression of the bias when the images are sampled below the Nyquist rate (and hence aliased) is much more complicated. To give an example of the complexity, we present the calculations for the case where the sampling rate is half the Nyquist rate. In other words, the DTFT of the sampled *aliased* signal denoted $F_a(\theta)$, is related to the original DTFT signal $F(\theta)$ according to

$$F_a(\theta) = \frac{1}{2}\left[F\left(\frac{\theta}{2}\right) + F\left(\frac{\theta - 2\pi}{2}\right)\right],\ \theta \in [-\pi, \pi] \tag{4.10}$$

From this, we can study the numerator and denominator of (4.8) for the case when aliasing is present. First, we study the numerator which looks like

$$\begin{aligned}
&\int_{-\pi}^{\pi} \left[\frac{j\theta}{2} F\left(\frac{\theta}{2}\right) + \frac{j(\theta-2\pi)}{2} F\left(\frac{\theta-2\pi}{2}\right)\right]^* \\
&\qquad \left[F\left(\frac{\theta}{2}\right)(1-e^{jv_0\frac{\theta}{2}}) + F\left(\frac{\theta-2\pi}{2}\right)(1-e^{jv_0\frac{\theta-2\pi}{2}})\right] d\theta \\
=\ & \int_{-\pi}^{\pi} |F(\theta)|^2 \theta \sin(v_0\theta) d\theta + \\
& \int_{-\pi}^{\pi} \mathrm{Re}\left[F(\frac{\theta}{2})F(\frac{\theta-2\pi}{2})\right]\left(\frac{\theta}{2}\sin\left(\frac{v(\theta-2\pi)}{2}\right) + \frac{\theta-2\pi}{2}\sin\left(\frac{v\theta}{2}\right)\right) d\theta
\end{aligned}$$

Thus, we see that the aliased component adds an additional term to the numerator of (4.8). Likewise, we see that the denominator of (4.8) is given by

$$\begin{aligned}
&\int_{-\pi}^{\pi} |\frac{\theta}{2} F\left(\frac{\theta}{2}\right) + \frac{\theta-2\pi}{2} F\left(\frac{\theta-2\pi}{2}\right)|^2 d\theta \\
&= \int_{-\pi}^{\pi} |F(\theta)|^2\theta^2 d\theta + \int_{-\pi}^{\pi} \mathrm{Re}\left[F(\frac{\theta}{2})F(\frac{\theta-2\pi}{2})\right] \frac{\theta(\theta-2\pi)}{2} d\theta
\end{aligned}$$

which is again a perturbation of the denominator for the non-aliased case. For the remainder of this chapter, however, we assume that the images are sampled above the Nyquist. We leave the analysis of the bias for the aliased scenario to future research.

4.1.2 Bias From Gradient Approximation

In the previous section, we assumed that the derivative values at the sample points were known prior to the estimation process. As mentioned previously, in most applications, the derivative information is not available. Another source of error in gradient-based estimation arises from the need to approximate the gradient or the derivatives of the signal $f(n)$. These gradients (derivatives) $f'(n)$ must be approximated from the measured data using a gradient filter $g(n)$ applied to one of the available images:

$$\tilde{f}'(n) \approx z_0(n) * g(n) = [f(n) + \epsilon_0(n)] * g(n) \tag{4.11}$$

(where $*$ represents convolution). It is common practice to apply pre-smoothing filters to each image prior to estimation. Using gradient filters, the form of (4.3) is now modified to be

$$z(n) \quad = \quad v_0\tilde{f}'(n) + \tilde{R}(n, v_0) + \epsilon(n). \tag{4.12}$$

As expected, the remainder term $\tilde{R}(n, v_0)$ plays a critical role in the overall estimator bias.

The error resulting from such derivative approximation has been noted before in the literature. For instance, in [55], the bias function was derived only for the case when f is a single sinusoid function. In addition, the works of [57] and [58] explored the effect of approximation errors in estimating the gradient for local estimation. Much of the analysis in these works, however, starts from the assumption that the optical flow model applies to the image sequence exactly, or that the remainder term is negligible. Specifically, in [58], the results qualitatively described estimator bias in terms of image spectral content and were based on overly simplified bias approximation by examining only the second order approximation error specifically for the forward difference gradient approximation. The authors in [57] note that the gradient approximation error increases as the image function exhibits higher energy in the second derivatives $f''(n)$. Using this observation, they propose an estimator post-processing scheme which examines the second order derivatives of the image and rejects specific estimates according to a thresholding scheme. Other works, such as [59], have noted that errors in the gradient approximation tend to produce biased estimates. In [59], however, it is assumed that these errors are completely random in nature and drawn from some simple distribution. They develop overly simplified statistical bias models based on these distributions for the gradient approximation errors. Recently, the work of [60] investigates a method for minimizing the bias associated with such random errors for an application in vehicle tracking. Instead of treating these errors as random, as we shall show, approximation errors resulting from deterministic systematic modelling error dominate the estimator bias for gradient-based estimators at SNRs found in typical imaging systems.

When we use the gradient approximations, the estimator (4.6) becomes

$$\begin{aligned}\hat{v}_0 &= \frac{\int_{-\pi}^{\pi} jG(\theta)Z_2(\theta)Z^*(\theta)d\theta}{\int_{-\pi}^{\pi} |G(\theta)Z_2(\theta)|^2 d\theta} \\ &= \frac{\int_{-\pi}^{\pi} jG(\theta)\left[F(\theta)+\xi_2(\theta)\right]Z^*(\theta)d\theta}{\int_{-\pi}^{\pi} |G(\theta)\left[F(\theta)+\xi_2(\theta)\right]|^2 d\theta}\end{aligned} \tag{4.13}$$

where $G(\theta)$ represents the DTFT of $g(n)$ and $\xi_2(\theta)$ represents the DTFT of the noise samples

$\epsilon_2(n)$. In general, the derivative filter $g(n)$ is usually a symmetric, linear-phase, FIR filter whose transform is $jG(\theta)$ where

$$G(\theta) = \sum_{i=1}^{\tau} c_i \sin(i\theta). \tag{4.14}$$

Such a filter is referred to as a $2\tau + 1$ tap filter. Unfortunately, taking the expectation of (4.13) is very difficult. To simplify the equation, we ignore the noise in the derivative approximation. As such, we can write the expected value of the estimator as

$$\begin{aligned} E[\hat{v}_0] &\approx E\left[\frac{\int_{-\pi}^{\pi} jG(\theta)\,[F(\theta)]\,Z^*(\theta)d\theta}{\int_{-\pi}^{\pi} |G(\theta)\,[F(\theta)]\,|^2 d\theta}\right] \\ &= \frac{\int_{-\pi}^{\pi} |[F(\theta)|^2 G(\theta)\sin(v_0\theta)d\theta}{\int_{-\pi}^{\pi} |G(\theta)\,[F(\theta)]\,|^2 d\theta} \end{aligned}$$

This assumption is quite reasonable at high SNR situations where basically we are examining the deterministic bias from modelling error as opposed to statistical error. In Section 4.3, we will show the SNR region where this model accurately describes estimator performance and demonstrate that this SNR region is typical for many imaging systems including commercial video cameras. Using this approximation, we see that the bias function is given by

$$b(v_0) = \frac{\int_{-\pi}^{\pi} |F(\theta)|^2 \left[G(\theta)\sin(v_0\theta) - vG^2(\theta)\right] d\theta}{\int_{-\pi}^{\pi} |G(\theta)F(\theta)|^2 d\theta}. \tag{4.15}$$

We can see here that the this equation differs from the original equation (4.9) only in that the exact derivative operator $j\theta$ is replaced by the approximate derivative kernel $jG(\theta)$.

To verify this approximation of the bias function, we measure the actual estimator bias using the gradient kernel $g(n) = [.1069\,.2846\;\;0\;-.2846\;-.1069]$ on the same function shown in Figure 4.1. This derivative kernel comes from [1]. The left graph of Figure 4.2 shows the results of the bias. The experimental bias again follows the bias predicted by (4.15) almost exactly. The measured bias functions shown in [55] also appear to follow this trend, providing further validation of our bias expression. Again, we note that the increased signal bandwidth produces increased estimator bias.

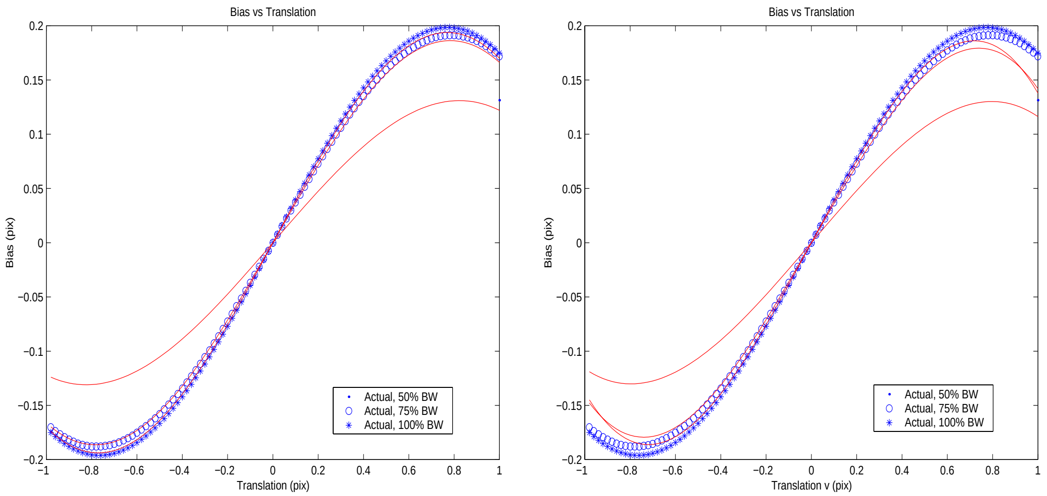

Figure 4.2: Plot of actual estimator bias and predicted bias (solid lines) from equation (4.15)-left graph and equation (4.19)-right graph

4.2 Analysis of Gradient-Based Estimator Bias

In this section we further explore the deterministic bias approximation (4.15). We will show how the structure of the bias function explains much of the heuristic knowledge about gradient-based estimators and suggests methodologies for improving performance. In particular, we will explore how the image spectrum, translation, and gradient kernel all affect the bias of the gradient-based estimator. In keeping with our original presentation, all of the analysis is presented for the 1-D scenario. Such analysis is also instructive as it is indicative of the bias of a projection-based algorithm presented in Chapter 2. Finally, we note that a simple comparison of the bias for the direct (2-D) estimation algorithm with that of a projection-based algorithm is presented in Appendix 4.B.

We begin by analyzing the bias function (4.9) wherein the exact derivatives are available to the estimator. To understand the bias, we expand the $\sin$ function in a Taylor series about $v = 0$ to get

$$b(v_0) = \frac{\int_{-\pi}^{\pi} |F(\theta)|^2 \left[v\Lambda_1(\theta) - v_0^3\Lambda_2(\theta) + v_0^5\Lambda_3(\theta)\ldots \right] d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2 \theta^2 d\theta}$$

where the terms of the sequence are $\Lambda_1(\theta) = 0$, $\Lambda_2(\theta) = \frac{\theta^4}{3!}$, $\Lambda_3(\theta) = \frac{\theta^6}{5!}$ and so on. Since the factorial in the denominator dominates these Λ functions, the coefficients of the Taylor approximation die off quickly. Only for very large translations, often larger than is found in typical registration problems, will these higher order terms affect the bias function. This suggests that for small v_0, the bias can be approximated as a cubic function of translation v_0 according to

$$b(v_0) \quad \approx \quad -\frac{v_0^3}{3!}\frac{\int_{-\pi}^{\pi}|F(\theta)|^2\theta^4 d\theta}{\int_{-\pi}^{\pi}|F(\theta)|^2\theta^2 d\theta}. \tag{4.16}$$

This coefficient ratio can be interpreted as the energy in the second derivative over the energy in the first derivative of $f(x)$. In general, the Taylor series can be explained in the spatial domain as

$$b(v_0) = -\frac{v_0^3}{3!}\frac{\sum[f''(n)]^2}{\sum[f'(n)]^2} + \frac{v_0^5}{5!}\frac{\sum[f^{(4)}(n)]^2}{\sum[f'(n)]^2} - \dots \tag{4.17}$$

Basically, these higher order terms depend on the smoothness of the function $f(x)$. For sufficiently smooth functions the energy in these higher derivatives is negligible, suggesting that the bias is well approximated by the cubic function given in (4.16). The accuracy of this bias approximation is evident in right graph of Figure 4.1.

We repeat this analysis for the more complete bias function (4.15), expanding the function in a Taylor series about $v = 0$ to produce

$$b(v_0) \quad = \quad \frac{\int_{-\pi}^{\pi}|F(\theta)|^2\left[v_0\tilde{\Lambda}_1(G,\theta) - v_0^3\tilde{\Lambda}_2(G,\theta) + v_0^5\tilde{\Lambda}_3(G,\theta)\dots\right]d\theta}{\int_{-\pi}^{\pi}|G(\theta)F(\theta)|^2 d\theta} \tag{4.18}$$

where the terms are of the sequence are $\tilde{\Lambda}_1(G,\theta) = \theta G(\theta) - G^2(\theta)$, $\tilde{\Lambda}_2(G,\theta) = \frac{\theta^3}{3!}G(\theta)$, $\tilde{\Lambda}_3(G,\theta) = \frac{\theta^5}{5!}G(\theta)$ and so on. From this approximation, we see that the polynomial coefficients depend on the relationship between the gradient kernel $G(\theta)$ and the image magnitude spectrum $|F(\theta)|$. Again, we simplify the bias expression by truncating the power series to that of a cubic function of v_0.

$$b(v_0) \quad \approx \quad v_0\left(\frac{\int_{-\pi}^{\pi}|F(\theta)|^2\tilde{\Lambda}_1(G,\theta)d\theta}{\int_{-\pi}^{\pi}|G(\theta)F(\theta)|^2 d\theta}\right) - v_0^3\left(\frac{\int_{-\pi}^{\pi}|F(\theta)|^2\tilde{\Lambda}_2(G,\theta)d\theta}{\int_{-\pi}^{\pi}|G(\theta)F(\theta)|^2 d\theta}\right) \tag{4.19}$$

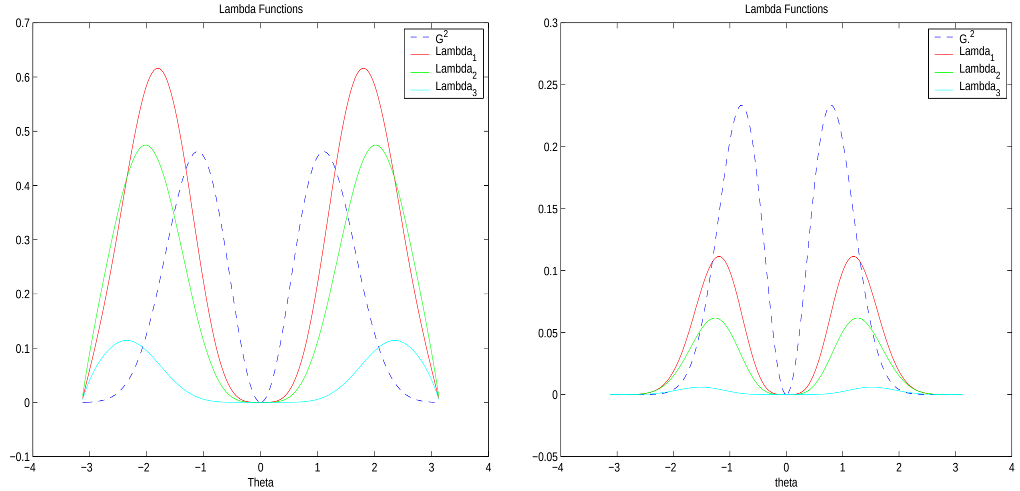

Figure 4.3: Original (left) and Filtered (right) versions of $\tilde{\Lambda}$ and $|G|^2$ functions. The filter function $h(n) = [0.035\ 0.248\ 0.432\ 0.248\ 0.035]$ is suggested in [1]

In the right graph of Figure 4.2, we show the same experimental bias curves as in the left graph of Figure 4.2, this time using the cubic approximation of (4.19). We see that the approximation is quite close for the sub-pixel region of v_0.

4.2.1 Bias and Image Spectrum

The spectrum of the image/function plays an important role in the bias expression (4.15). One way to shape the image spectrum is through the use of image filters. For instance, it is well-known that pre-smoothing the images prior to estimation improves the performance of the gradient-based estimators [14], [1]. This pre-smoothing operation takes the form of a low-pass filter $H(\theta)$. To understand this, in the left graph of Figure 4.3 we plot the Λ functions found in (4.18), again using the gradient kernel $G(\theta)$ from [1].

Basically, the $\tilde{\Lambda}$ functions and the $|G|^2$ (where $|G|^2 = |G(\theta)|^2$) term control the numerator and denominator of the coefficients of the bias polynomial of (4.18). Looking at the left graph of Figure 4.3, we see that the $|G|^2$ term is larger than all of the $\tilde{\Lambda}$ functions up to the frequency of about $\frac{\pi}{3}$ for $\tilde{\Lambda}_1$, $\frac{\pi}{2}$ for $\tilde{\Lambda}_2$ and about $\frac{2\pi}{3}$ for $\tilde{\Lambda}_3$. If the spectrum of the function were bandlimited such that the image contained no spectral energy outside these frequencies, we know that the coefficients of the bias function would be less than 1. Beyond these critical frequencies the numerator $\tilde{\Lambda}$ functions weight the spectrum more heavily than the denominator $|G|^2$ function, which has the effect of increasing the bias coefficients. As we will show, this explains the well-known assertion that pre-smoothing the images improves estimator performance. In addition to removing noise, the image pre-smoothing has the effect of minimizing the high frequency spectral components, thereby minimizing the polynomial coefficients. Furthermore, since higher order terms place more emphasis on the high frequency information than the lower order terms, the pre-smoothing also has the effect of reducing the influence of the higher order terms.

For instance, the authors in [1] suggest using a 5-tap pre-smoothing low-pass filter $h(n)$. Effectively, this pre-smoothing changes the weighting functions into $|GH|^2$, $\tilde{\Lambda}_1|H|^2$, $\tilde{\Lambda}_2|H|^2$ and so on. In Figure 4.3 we also show the filtered versions of the Λ functions. Unlike the original $\tilde{\Lambda}$ functions, the smoothed versions have much smaller magnitude than the $|G|^2$ function and very small regions wherein the numerators would weight the spectrum F more than the denominator. This phenomenon tends to minimize the bias polynomial coefficients. For high SNR situations where the bias dominates MSE, pre-smoothing tends to minimize the bias in general. This is shown in Figure 4.4, where the bias is plotted as a function of translation wherein the function in Figure 4.1 is filtered by different pre-smoothing filters. Each of the filters was a Gaussian kernel with 10 taps where the low-pass cutoff frequency was controlled by the standard deviation (SD) of the Gaussian. These low-pass filters were not designed in any optimal fashion, and yet we still see a significant reduction in bias. For this experiment, we extended the range of translation beyond subpixel translation to show the dramatic improvement for larger values of v_0.

Pre-smoothing an image also has the benefit of averaging, essentially decreasing the variance of the noise. Again, this pre-smoothing would, however, decrease the Fisher Information by reducing the effective bandwidth of the signal. Interestingly, one could pose an

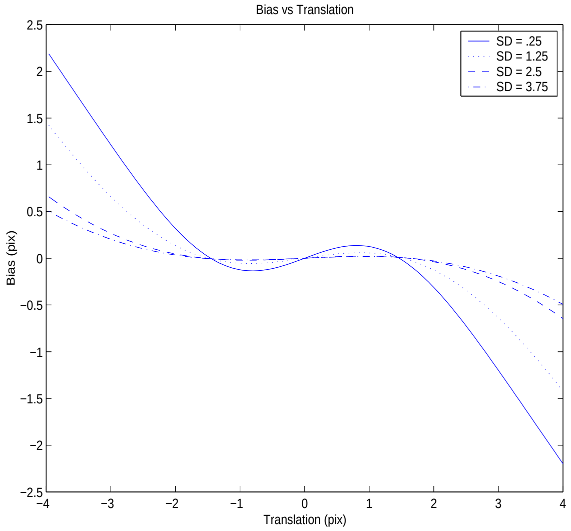

Figure 4.4: Bias vs translation for different pre-filters.

optimization problem of finding the pre-filter $H(\theta)$ that minimizes the bias in a sense similar to [61]. Of course, this optimization would only make sense for very high SNR situations since pre-smoothing the image tends to minimize the FIM, thereby making the estimator more sensitive to noise. We leave this interesting problem for future work.

4.2.2 Bias and Gradient Kernel

Another important ingredient in the bias function is the choice of gradient filters $G(\theta)$. The gradient kernel defines the shape of the Λ functions which in turn controls the bias coefficients. The left graph of Figure 4.5 exhibits the performance in estimating translation using the three filters, and also the bias when the exact derivative were used. The experimental setup was similar to previous experiments wherein the function used was shown in Figure 4.1 and no noise was added to simulate infinite SNR.

Examining the bias curves, it might appear that the Nestares/Heeger filter minimizes the bias, even producing better estimates than when the exact derivatives were known prior to estimation. In the right graph of Figure 4.5 we examine the curves more closely in the range

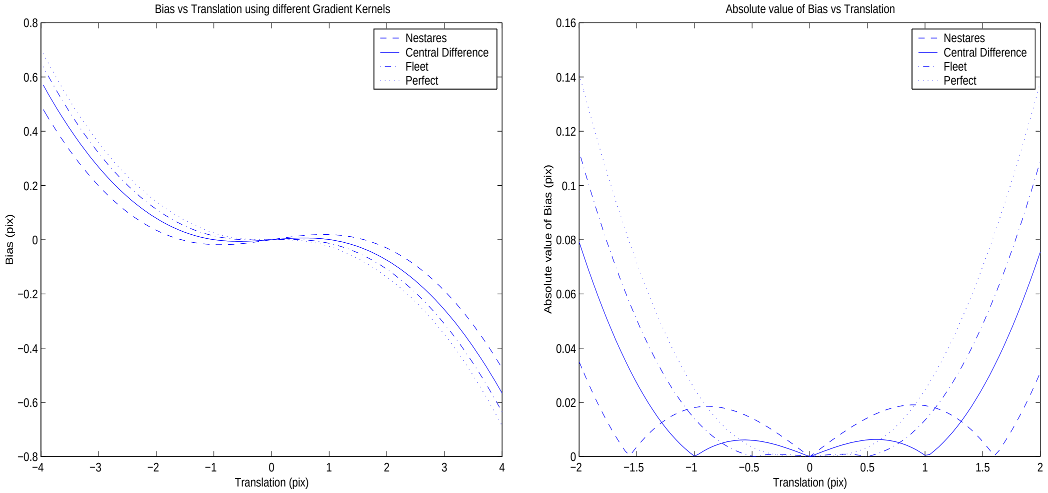

Figure 4.5: Bias vs translation for different gradient filters.

$v_0 \in [-2, 2]$, and display absolute value of the bias. In the subpixel range ($v_0 \in [-1, 1]$), we see that the Nestares/Heeger filer, in fact, produces estimators with largest bias magnitude.

We see from these plots that there is a tradeoff in performance in estimating large and small translations. It appears that the tradeoff concerns the linear term in the bias polynomial approximation. The central difference and Fleet derivative filters are the 2nd and the 4th order optimal approximations to the infinite ordered ideal derivative filter. Thus, these filters produce derivative estimates closer to the exact derivative than the filter of Nestares/Heeger. This more accurate derivative approximation tends to minimize the linear term of the bias polynomial leaving basically the cubic term as in the case of (4.16). The filter of Nestares/Heeger, however, is not an approximation to the ideal derivative filter. As such, it has a much larger linear coefficient. This larger linear coefficient explains its poor performance around the subpixel range, and yet it produces a linear improvement for larger translations. Again, this phenomenon suggests a certain optimization framework similar to [61] where the gradient kernel may be optimized over some range of translations. We will address this idea later in the chapter.

4.2.3 Bias and Translation

Finally, we examine how the bias varies with the unknown translation v_0. As expected, the first order approximation used to generate the linear gradient-based estimator is accurate only for small translations. Thus, with perfect knowledge of the image derivatives, the magnitude of the bias tends to increase with the translation and the estimates are always biased towards zero, or underestimated. When the derivatives are only approximated using a gradient kernel, however, there are essentially two regions of operation wherein the estimates could be overestimated and underestimated. These regions are easy to identify when examining the cubic approximation of the bias (4.19). Setting (4.19) equal to zeros, the resulting roots of the cubic polynomial are

$$\tilde{v}_0 = 0, \ \pm \left(\frac{\int_{-\pi}^{\pi} |F(\theta)|^2 \Lambda_1(G,\theta) d\theta}{\int_{-\pi}^{\pi} |F(\theta)|^2 \Lambda_2(G,\theta) d\theta} \right)^{\frac{1}{2}} . \tag{4.20}$$

Instead of biasing the estimates towards 0 as in the case where the derivatives were known exactly, the estimator produces estimates that are biased towards $\pm\tilde{v}_0$. Examination of the bias in the right graph of Figure 4.5 shows that these values are around $\tilde{v}_0 = 1.5$ for Nestares/Heeger, $\tilde{v}_0 = 1$ for the central difference, and $\tilde{v}_0 = .5$ for the Fleet gradient filters. In fact, we find that these value of $\tilde{v}_0$ do not vary much across different images, for any reasonable derivative filter.

Whichever gradient kernel is used, if the kernel approximates the derivative, the magnitude of the bias will tend to worsen for values of $|v_0| > |\tilde{v}_0|$. In fact, the cubic approximation of bias suggests that even the relative bias $\frac{b(v_0)}{v_0}$ increases as a quadratic function of v_0. This partly explains the success of multiscale gradient-based methods in estimating large translations. The multiscale pyramids are constructed through a process of low-pass filtering and downsampling. We have already shown how the low-pass filtering improves estimator performance. The downsampling reduces the magnitude of the translation by the downsampling factor, the common factor being 2. Using this downsampling factor, the translation to be estimated at the lth level of the pyramid becomes $v_0^l = \frac{v_0}{2^l}$. This synthetic reduction in translation magnitude allows for estimation with smaller relative bias. The reduction in bias is most effective when the unknown translation is greater than a few pixels. In this case, the downsampling maps the translation into a range of reasonably small bias. In practice, the height of the pyramid L is designed such that the expected downsampled velocity at the coarsest level is in $v_0^L \in [-2, 2]$ pixels/frame where the magnitude of the relative bias is not very large.

The iterative nature of the multiscale pyramid raises an important question concerning the general convergence of iterative gradient-based estimators. Iterative methods for gradient-based estimation have been used to improve performance [37], [1], [42]. These methods work by iteratively estimating motion, *undoing* this estimated motion, and estimating the residual motion not captured by the previous estimate. At very high SNR, the residual motion is dominated by the estimator bias. In practice, different methods are used to *undo* the previously estimated motion, often relying on some warping/resampling scheme. We would like to know if these iterative methods will converge, and if so, whether they will converge to an unbiased estimate of v_0.

To simplify the analysis, we assume that the warping methods work perfectly to synthesize a shifted version of the images [1]. In fact, we see that the error in the gradient approximation could lead to oscillatory instability in the iterative gradient-based estimator. To see this, assume that an initial estimate of translation using the gradient based estimator was given by $\hat{v}_0^0 = v_0 + b(v_0)$ (where superscript 0 indicates the iteration number). After *perfect* warping, the residual translation would simply be $r = -b(v_0)$. The estimate of this residual motion will be $\hat{r} = -b(v_0) + b(-b(v_0))$ such that the updated motion estimate becomes $\hat{v}_0^1 = \hat{v}_0^0 + \hat{r}_0 = v_0 + b(-b(v_0))$. Thus, if $|b(v_0)| < |v_0|$ for all v_0, then $|b(-b(v_0))| < |b(v_0)|$ and so on, suggesting convergence to an unbiased estimate. Practically speaking, we are only interested in this relationship for very small v_0 since the residual motions are often within the range $[-\tilde{v}_0, \tilde{v}_0]$. In this region, we use the cubic approximation of (4.19) represented as

$$b(v_0) = \frac{\gamma_1}{\gamma_2} v_0 - \frac{\gamma_3}{\gamma_2} v_0^3 \qquad (4.21)$$

where the γ variables represent the numerator and denominators of the polynomial bias ap-

[1] unlikely given the ill-posed nature of image resampling

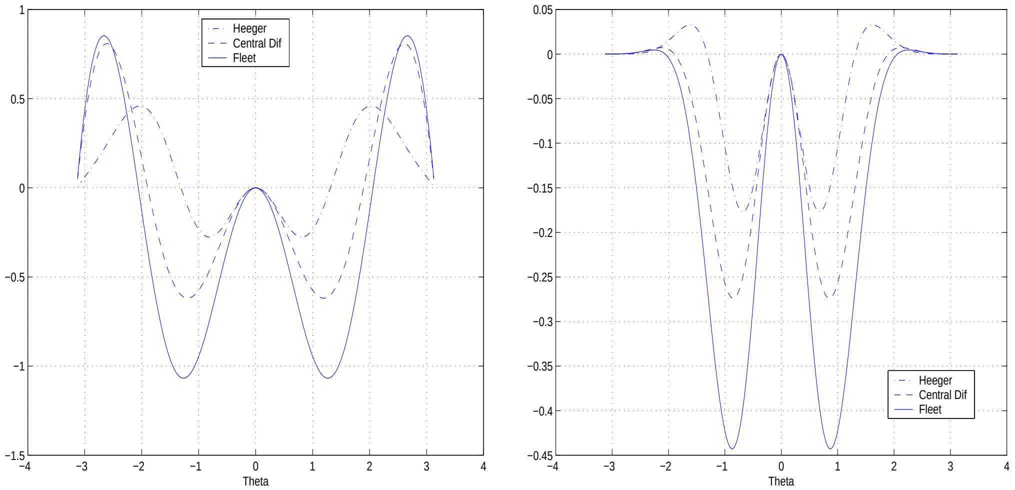

Figure 4.6: Original (left) and filtered (right) plot of $\theta G(\theta) - 2G^2(\theta)$.

proximation. Because of the symmetry of the bias function, we must examine whether or not $|b(v_0)| < |v_0|$ for all $v_0 \in [0, \tilde{v}_0]$. In this region, we see that the condition $|b(v_0)| < |v_0|$ will be satisfied if

$$\frac{|\gamma_1|}{\gamma_2} \leq 1. \tag{4.22}$$

Furthermore, it can be shown that under the very general assumption that the filter $G(\theta)$ is in fact a derivative-type operator, we have $\gamma_1 \geq 0$. Thus, if $\gamma_2 \geq \gamma_1$, then we can safely assume that $|b(v_0)| < |v_0|$ for small translations assuring that the iterative method will converge to an unbiased estimate since the bias is reduced at every iteration. However, if $\gamma_2 < \gamma_1$ then the estimator will oscillate between $\hat{v}_0 = v_0 \pm v_0^*$.

Since the condition of convergence depends on

$$\gamma_1 - \gamma_2 = \int_{-\pi}^{\pi} |F(\theta)|^2 \left[\theta G(\theta) - 2G^2(\theta)\right] d\theta, \tag{4.23}$$

we plot $\theta G(\theta) - 2G^2(\theta)$ in left graph of Figure 4.6. For the iterative estimator to converge, most of the spectral energy must be located in the low frequency range where the weighting function $\theta G(\theta) - 2G^2(\theta)$ applies negative weight. If too much high frequency content is present, the difference $\gamma_1 - \gamma_2$ will be positive and the algorithm will not converge to an unbiased estimate.

Pre-smoothing the image minimizes the likelihood that $\gamma_1 - \gamma_2 > 0$ since most of the weighting function $\theta G(\theta) - 2G^2(\theta)$ is negative. Although multiscale iterative methods significantly decrease estimator bias in practice as evidenced in Figure 3.6, they still may contain estimator bias.

4.3 MSE Performance of the Gradient-Based method

Armed with an approximate expression for the bias function, we can now examine the full performance bound given by (3.1) for the gradient-based estimators. In examining this bound, we find that the bias dominates the MSE performance for typical imaging systems with high SNR. Finally, we show experimental evidence justifying a general rule-of-thumb for performance of 2-D gradient-based image registration.

In order to use the performance bound given by (3.1), we must first examine the derivative of the bias function. Using the bias expression (4.15) we see that

$$b'(v_0) + 1 = \frac{\int_{-\pi}^{\pi} |F(\theta)|^2 G(\theta)\theta \cos(v_0\theta) d\theta}{\int_{-\pi}^{\pi} |G(\theta)F(\theta)|^2 d\theta}. \tag{4.24}$$

Using these expressions, we see that the complete MSE performance bound is given by

$$\begin{aligned} MSE(v_0) &\geq \frac{[b'(v_0)+1]^2}{J(v_0)} + b^2(v_0) \\ &= J^{-1} \frac{\left(\int_{-\pi}^{\pi} |F(\theta)|^2 G(\theta)\theta \cos(v_0\theta) d\theta\right)^2}{\left(\int_{-\pi}^{\pi} |G(\theta)F(\theta)|^2 d\theta\right)^2} + \\ &\quad \frac{\left(\int_{-\pi}^{\pi} |F(\theta)|^2 \left[G(\theta)\sin(v_0\theta) - v_0 G^2(\theta)\right] d\theta\right)^2}{\left(\int_{-\pi}^{\pi} |G(\theta)F(\theta)|^2 d\theta\right)^2} \end{aligned} \tag{4.25}$$

where the Fisher Information is independent of v_0 and is given by

$$J = \frac{1}{\sigma^2} \int_{-\pi}^{\pi} |F(\theta)|^2 \theta^2 d\theta. \tag{4.26}$$

In practice, we calculate the Fisher Information using derivative approximations.

Here we conduct a Monte-Carlo (MC) simulation to verify the accuracy of our complete MSE bound. Ideally, at high SNR the complete bound given by (4.25) predicts actual estimator performance. We construct a bandlimited signal using

$$f(n) = \sum_{i=1}^{D} \frac{1}{i} \sin\left(\frac{\pi n i}{100} - \phi_i\right), n = 1\dots 100 \tag{4.27}$$

where ϕ_i is a fixed phase generated by drawing from a uniform distribution. We chose to use a closed-form expression for f so that that the *exact* values of the function derivative are available. These derivatives were used to calculate the exact FIM used in the complete CR bound of (4.25). Actual estimator performance is measured by performing 500 MC runs at each value of SNR and averaging the error. The gradient kernel used was the Nestares filter from [1].

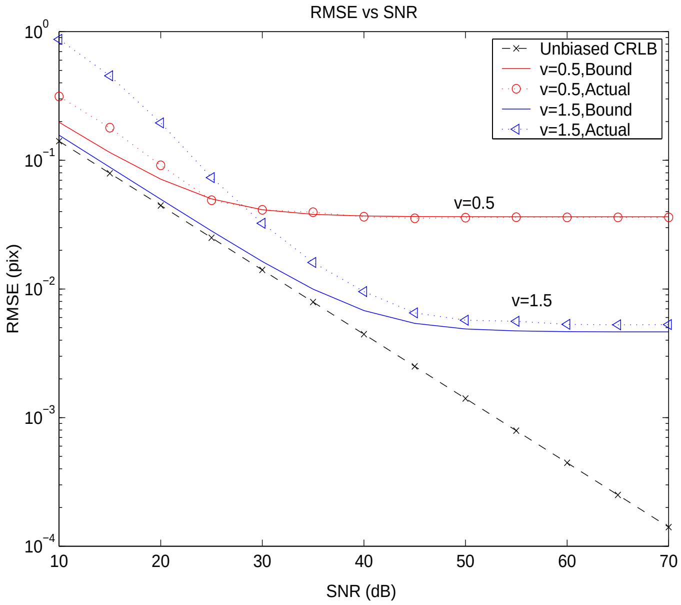

Figure 4.7: Experimental RMSE and the corresponding complete CR bound vs SNR.

The results of the simulation are shown in Figure 4.7 which compares the RMSE for the gradient based estimator with both the unbiased CR bound (3.2), and the full bound (4.25). The actual estimator performance seems very close to the performance bound predicted by (4.25) at high SNR. This verifies that the bias function given by (4.15) is in fact accurate. For low SNR, however, both bounds are overly optimistic. This could be due in part to the approximation made in obtaining the simplified bias function. In general, nonlinear estimation problems suffer from what is known as the threshold effect [35]. This threshold effect is characterized by a significant departure from the CR bound as the SNR degrades.

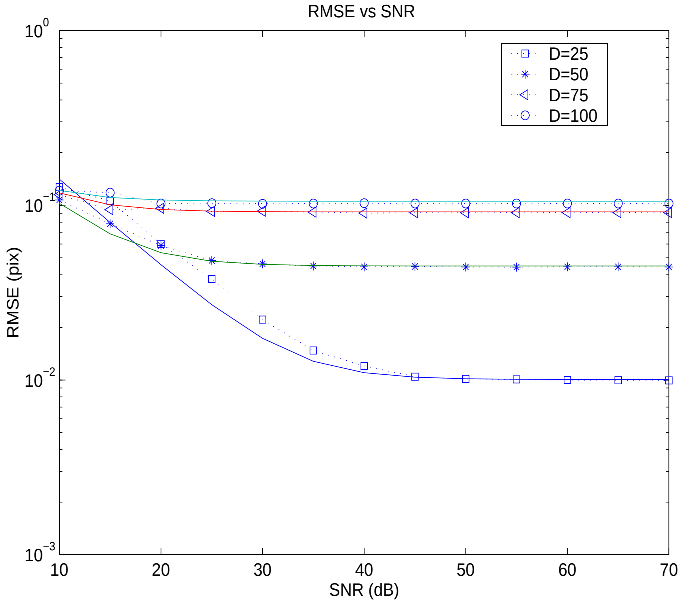

Figure 4.8: Experimental RMSE and corresponding complete CR bound vs SNR as it relates to signal bandwidth.

To understand the relationship between bandwidth and performance bound, we plot the expected performance bound for $v_0 = 0.1$ for different values of D in (4.27), (which essentially encodes the bandwidth in the definition of f) in Figure 4.8. This figure shows the tradeoff between bias and variance as it relates to image bandwidth where D is the percentage of full bandwidth. As mentioned before, energy in higher frequencies tends to increase the Fisher Information, thereby improving estimator variance, but tends to worsen the effect of bias. Overall, it is apparent that bias dominates the MSE for images with much high energy in the high spatial frequencies.

Lastly, we extend this complete MSE performance bound for the case of 2-D image registration. The equations for 2-D bias can be found in Appendix 4.A. To provide a rule of thumb value for expected estimator performance, we use the following performance measure.

$$\Delta_{v0_1}\Delta_{v0_2}\sum_{v0_1}\sum_{v0_2}\overline{rmse}(\mathbf{v}_0) \tag{4.28}$$

where Δ_{v0_i} defines the sampling grid over the space of translations. This provides a measure of the average magnitude error over a range of unknown translations. The corresponding CR bound used to compare actual estimator performance is given by

$$\frac{1}{4V_1V_2}\int_{-V_1}^{V_1}\int_{-V_2}^{V_2}T(\mathbf{v}_0)dv_{0_1}dv_{v_2} \quad . \tag{4.29}$$

For our experiment, we examine estimator performance for sub-pixel estimation where $\mathbf{v}_0 \in [-1,1]\times[-1,1]$. The tree image was again shifted synthetically as before using the method of [55]. For each value of SNR, 500 MC runs were performed and averaged to obtain the MSE matrices. To evaluate the improvement using image pre-smoothing, we apply a 9-tap Gaussian filter with standard deviation of 1 and 2 pixels. To compute the MSE bound, we estimate the spectrum $F(\theta_1,\theta_2)$ using the DFT coefficients of the clean Tree image. To take into account the noise reduction resulting from image pre-smoothing, we modified noise variance used to compute the FIM by $\tilde{\sigma}^2 = \frac{\sigma^2}{\sum h^2(n)}$ where $h(n)$ are the coefficients of the Gaussian filter. Again, the gradient filter used was from [1]. Figure 4.9 shows the performance predicted by (4.29) and actual experimental performance of (4.28) from the Monte-Carlo experiments using the Tree image as the base signal.

The performance bound appears to be a good predictor of actual estimator performance at high SNR situations. The estimator performance for SNR's at about 20-40 dB shows unexpected improvement over the high SNR situation. Most likely, this results from the statistical bias present in the estimator for low SNR situations. It was shown in [59] and [60] that the statistical bias for noisy images tends to produce underestimates of translation or negative bias. Since the deterministic bias using the [1] filter is positive for subpixel motion, we deduce that these two biases tend to cancel each other out, thereby improving performance at low SNR. As significant low-pass filtering is applied to the image, estimator performance improves dramatically. Basically, the deterministic bias again dominates estimator bias and we have predictably

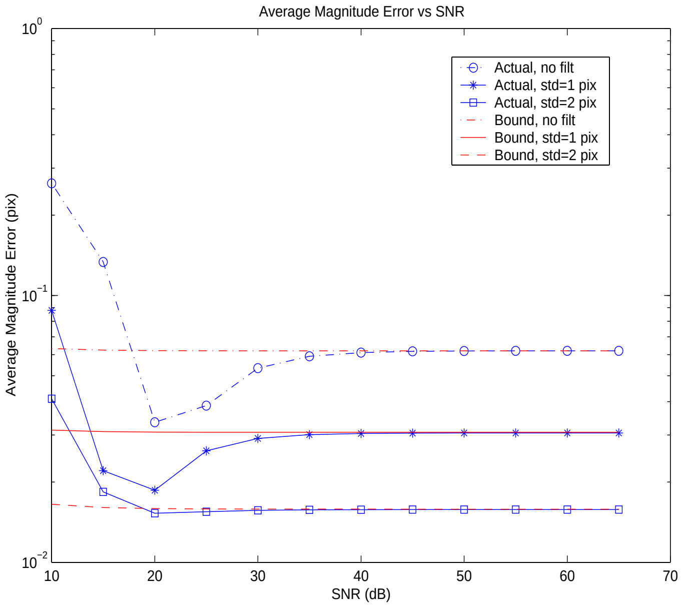

Figure 4.9: Predicted and measured average performance measured by (4.28)

improved estimator performance. This experiment presents the possibility of subpixel image registration accuracy down to almost one hundredth of a pixel for the gradient-based estimator under the ideal situation when the image is known to be sampled above the Nyquist rate. Again, this experiment correlates well with the results shown in Figure 3.6 of Chapter 3. Thus, we can expect a rule of thumb performance bound limiting the performance of image registration under ideal situations to an accuracy of over one hundredth of a pixel for non iterative gradient-based estimation.

4.4 Filter Design for Gradient-Based Motion Estimation

In the previous section we characterized the bias associated with gradient-based estimators for high SNR situations. Much of this estimator bias is dependent on the choice of gradient filters used during the estimation process. Very little work has been done addressing the design of filters specifically for application to the problem of motion estimation. To our knowledge, such an approach was first studied in [61] which extends the generic (not necessarily application specific) gradient filter design principles of [62]. In this section, we briefly review previous work in the area of filter design for motion estimation.

To begin, we modify the previous gradient-based forward model to take into account the use of a pre-smoothing filter $h(n)$ prior to estimation. We modify (4.12) to include these pre-smoothing filters as

$$\begin{aligned} \tilde{z}(n) &= h(n) * z_1(n) - h(n) * z_0(n) \\ &= v_0 \tilde{f}'(n) + \tilde{R}(n, v_0) + \tilde{\epsilon}(n). \end{aligned} \tag{4.30}$$

The work of [62] suggested that the filters should be designed to match the actual derivative of a reconstructed continuous function. Such a design philosophy attempts to minimize the approximation error associated with the use of FIR filters to estimate image gradients. The error is minimized assuming the image has the spectral amplitude of a *natural* image or $|F(\theta)| \approx \frac{1}{|\theta|}$. With these assumptions in place, [62] propose a cost function to design the filters $h(n)$ and $g(n)$. The cost function is expressed in the Fourier domain as

$$\mathcal{C}_1(\mathbf{h}, \mathbf{g}) = \int_{-\pi}^{\pi} \frac{1}{|\theta|} \left[j\theta H(\theta) - G(\theta) \right]^2 d\theta \tag{4.31}$$

where $H(\theta)$ and $G(\theta)$ are the Fourier transforms for the desired filters given by

$$\begin{aligned} H(\theta) &= \{\mathbf{h}\}_0 + 2\sum_{i=1}^{\tau_h} \{\mathbf{h}\}_i \cos(i\theta) \\ G(\theta) &= 2\sum_{i=1}^{\tau_g} \{\mathbf{g}\}_i \sin(i\theta). \end{aligned}$$

In [62], the solution was found by formulating the optimization problem as an eigenvalue problem. While [62] did not directly address the application of such filters to estimate motion, the filters have been noted to improve estimator performance [58].

As noted in [61], such a design procedure can also be used to find optimal filter coefficients both for the gradient filter and for a *pair* of pre-smoothing filters to be applied to gradient-based motion estimation. To do so, the form of (4.30) is generalized to take into account the application of distinct pre-smoothing filters to each image as in

$$\begin{aligned} \tilde{z}(n) &= h^1(n) * z_1(n) - h^2(n) * z_0(n) \\ &= v_0 \tilde{f}'(n) + \tilde{R}(n, v_0) + \tilde{\epsilon}(n) \end{aligned} \tag{4.32}$$

where $h^1(k)$, $h^2(k)$ are both linear phase FIR filter kernels. The filter coefficients are represented using vector notation as $\mathbf{g}, \mathbf{h}^1, \mathbf{h}^2$.

Using the more general formulation of (4.30), the authors of [61] derive a cost function taking into account a specific image as well as a range of possible translations $v_0 \in [-V, V]$. This cost function has the form

$$\begin{aligned} \mathcal{C}_2(\mathbf{h}^1, \mathbf{h}^2, \mathbf{g}) &= \int_{-V}^{V} \int_{-\pi}^{\pi} |F(\theta)|^2 \left| e^{j\theta v_0} H^1(\theta) - H^2(\theta) - v_0 G(\theta) \right|^2 d\theta dv_0 \\ &= \int_{-V}^{V} \int_{-\pi}^{\pi} |F(\theta)|^2 \left|\Upsilon(\theta)\right|^2 d\theta dv_0. \end{aligned} \tag{4.33}$$

We know heuristically that the filter should be designed to minimize the energy in the modelling error $\tilde{R}(n, v_0)$ weighted by the image spectrum over a range of unknown translations. The authors note that minimizing the error alone will not provide good filters, since the optimization tends to create ”non-informative” filters which contain most of their spectral energy at frequencies where the image spectral energy $F(\theta)$ is lowest. They correct this by adding an additional penalty term balancing the desire to optimize the filter for the given image with the desire to optimize the filter for an image with a flat spectrum ($F(\theta) = 1$). This modified cost function looks like

$$\mathcal{C}_2(\mathbf{h}^1, \mathbf{h}^2, \mathbf{g}) = \int_{-V}^{V} \int_{-\pi}^{\pi} \left[\alpha + (1-\alpha)|F(\theta)|^2\right] \left|\Upsilon(\theta)\right|^2 d\theta dv_0$$

where α is a tuning parameter to be applied during the filter design process. The authors also find a solution to this problem by again solving an eigenvalue problem. The paper provides experimental results displaying the advantage of using such image-adapted filters.

While these previous works have made fundamental contributions to gradient-based motion estimation, they ignore the the particular structure of the gradient-based motion estimator that ultimately characterizes the statistical performance of such estimators. In this chapter, we use the statistical performance of the estimator to guide the design process. Specifically, we present a scheme for designing filters which minimize the bias of the gradient-based image registration algorithm.

4.4.1 Designing Bias-Minimizing Filters

As we have shown, the general gradient-based motion estimators have significant estimator bias. In the previous section, we verified our bias expressions for high SNR scenarios. For many computer vision and image registration applications, the effective SNR of the imaging system falls into this high SNR regime.

In (4.15) we see that the bias depends on three factors: the image content f, the choice of filters $g(n)$ and $h(n)$, and the unknown translation v_0. Again, using the assumption that translation is limited to (and equally likely to be in) some range $v_0 \in [-V, V]$, we construct the following cost function for a particular image for finding filter coefficients:

$$\mathcal{C}(\mathbf{g}, \mathbf{h}) \;=\; \int_{-V}^{V} b^2(\mathbf{g}, \mathbf{h}) dv_0. \tag{4.34}$$

Such a cost function captures the desired goal of minimally biased estimates of image translation. Note that a statistical prior on v_0 could be incorporated into the integral of (4.34).

We now explore a simple method for minimizing such a cost function. Because of the complex nonlinear relationship between $\mathbf{g}$ and $\mathbf{h}$ in (4.34), we focus on the design of only the gradient filter coefficients $\mathbf{g}$. It would be possible to efficiently minimize (4.34) in a cyclic coordinated descent type algorithm which alternates between optimizing over $\mathbf{g}$ and $\mathbf{h}$. However, in practice, we have found that optimizing both filters offers only modest improvement in performance over optimizing the gradient filter alone. To a larger extent, the estimator bias depends on the choice of gradient filters. Thus, to save on computational resources, we suggest optimizing only the gradient filter. We note that the same simplifying steps used here to optimize the gradient filter can also be applied to optimizing the pre-smoothing filter as well. Here we present the algebraic simplifications useful for highly efficient filter optimization. Basically, we find the closed form solution to the integral associated with (4.34), allowing us to perform the optimization with minimal computational cost. We note that similar simplifying operations are applicable for the 2-D case as well.

Because we assume the signal is bandlimited and periodic, our Fourier transform $F(\theta)$ has only N terms where the spatial frequency is indexed by $\theta_i = \frac{i2\pi}{N}$, $i = 1 \dots N$. As such, we rewrite the bias function (4.15) in vector form as

$$b(v_0) = \frac{\mathbf{s}(v_0)^T \mathbf{KTg}}{\mathbf{g}^T \mathbf{T}^T \mathbf{KTg}} - v_0 \tag{4.35}$$

where

$$\begin{aligned}
[\mathbf{s}(v_0)]_i &= \sin(v_0\theta_i) \\
[\mathbf{K}]_{i,j} &= \begin{cases} |F(\theta_i)|^2|H(\theta_i)|^2, & i = j \\ 0, & i \neq j \end{cases} \\
[\mathbf{T}]_{i,j} &= \sin(j\theta_i).
\end{aligned}$$

In these equations, the i enumerates the spatial frequencies used in the DFT For instance, $\theta_i = \pi - \frac{i2\pi}{N}$.

The cost function $\mathcal{C}(\mathbf{g})$ can be written in vector form as

$$\begin{aligned}
\mathcal{C}(\mathbf{g}) &= \int_{-V}^{V} b^2(\mathbf{g}) dv_0 \\
&= \int_{-V}^{V} \left[v_0^2 + \frac{\mathbf{g}^T \mathbf{T}^T \mathbf{K}^T \mathbf{s}(v_0)\mathbf{s}(v_0)^T \mathbf{KTg}}{(\mathbf{g}^T \mathbf{T}^T \mathbf{KTg})^2} - 2v_0 \frac{\mathbf{s}(v_0)^T \mathbf{KTg}}{\mathbf{g}^T \mathbf{T}^T \mathbf{KTg}} \right] dv_0 \\
&= \frac{2V^3}{3} + \frac{\mathbf{g}^T \mathbf{T}^T \mathbf{K}^T \tilde{\mathbf{S}} \mathbf{KTg}}{(\mathbf{g}^T \mathbf{T}^T \mathbf{FTg})^2} - 2 \frac{\mathbf{p}^T \mathbf{FTg}}{\mathbf{g}^T \mathbf{T}^T \mathbf{FTg}} \qquad (4.36)
\end{aligned}$$

where

$$\begin{aligned}
\{\tilde{\mathbf{S}}\}_{i,j} &= \int_{-V}^{V} \sin(v_0\theta_i)\sin(v_0\theta_j) dv_0 \\
&= \frac{2\sin(V(\theta_i - \theta_j))}{\theta_i - \theta_j} - \frac{2\sin(V(\theta_i + \theta_j))}{\theta_i + \theta_j} \qquad (4.37)
\end{aligned}$$

and

$$\{\mathbf{p}\}_i = \int_{-V}^{V} v_0 \sin(\theta_i) dv_0 = \frac{2\sin(V\theta_i) - 2V\theta_i\cos(V\theta_i)}{\theta_i^2}.$$

It is the simple closed form solutions for such integrals that make such an optimization simple to implement. While not obvious, it is important to note that matrix $\tilde{\mathbf{S}}$ in (4.37) represents a convolution operation because of the spectral symmetry of $|F(\theta)H(\theta)|^2G(\theta)$ about $\theta = 0$. Thus, the left-multiply by the matrix $\tilde{\mathbf{S}}$ can be implemented using FFT operations, thereby removing the necessity of constructing the large matrix $\tilde{\mathbf{S}}$. In fact, none of the above computations are performed by explicitly constructing the matrices. This saves space and improves numerical efficiency. Such implementation details become critical for the 2-D scenario where matrix $\tilde{\mathbf{S}}$ becomes a dense $N^2 \times N^2$ matrix, effectively precluding explicit construction of the matrices.

Using these terms, we rewrite (4.36) as

$$\mathcal{C}(\mathbf{g}) = \frac{2V^3}{3} + \frac{\mathbf{g}^T\mathbf{\Gamma}_1\mathbf{g}}{(\mathbf{g}^T\mathbf{\Gamma}_2\mathbf{g})^2} - \frac{2\mathbf{q}^T\mathbf{g}}{\mathbf{g}^T\mathbf{\Gamma}_2\mathbf{g}} \tag{4.38}$$

where the terms

$$\begin{aligned} \mathbf{\Gamma}_1 &= \mathbf{T}^T\mathbf{K}^T\tilde{\mathbf{S}}\mathbf{K}\mathbf{T} \\ \mathbf{\Gamma}_2 &= \mathbf{T}^T\mathbf{F}\mathbf{T} \\ \mathbf{q} &= \mathbf{p}^T\mathbf{F}\mathbf{T} \end{aligned}$$

need to be computed only once during the optimization, greatly simplifying the overall computational complexity. The nonlinearity of the cost function becomes immediately apparent in the form of the cost function. Because the dimensions of the filters are relatively small (2-4 unique coefficients), we perform the filter design utilizing the black box Matlab optimization routine `fminunc`. This optimization takes only fractions of a second to run. In our experiments, we use a standard filter such as the filter of [1] as an initial guess for the optimization routine. While such an optimization routine does not guarantee a global minimum, we have found in practice that using such an optimization routine generates filters with improved performance.

4.4.2 Filter Design for 2-D Multiscale Iterative Registration

One important departure of our proposed method from the filter design methods of [62] and [61], is the extension to the design of 2-D filters. Both of these previous methods have addressed only the filter design problem for the 1-D case. The extension to the 2-D case involves designing generic 1-D filters. For example, in [61] it was recommended that 1-D filters be designed using $\alpha = 1$ (not image dependent) and applied to a 2-D image in a separable fashion.

In our case, we continue to assume that the 2-D filters are separable (although this is not a necessary assumption) to simplify not only the optimization routine, but also the application of such filters. However, unlike the previous works, our filters are designed taking into account the 2-D image spectral content. In this section, we show how the design of 2-D filters is a natural extension of the 1-D case presented in the previous section. In addition, we propose a methodology for designing filters for multiscale iterative image registration.

Filter Design for 2-D Registration

As in the 1-D case, we use a cost function of the form

$$\mathcal{C}(\mathbf{g}_1, \mathbf{g}_2) \quad = \quad \int \mathbf{b}(\mathbf{v}_0)^T \mathbf{b}(\mathbf{v}_0) d\mathbf{v} \tag{4.39}$$

to design our pair of gradient filters $\mathbf{g}_1$ and $\mathbf{g}_2$. While somewhat tedious to present, the same algebraic simplifications apply to $\mathcal{C}(\mathbf{g}_1, \mathbf{g}_2)$ as those shown in Section 4.4.1.

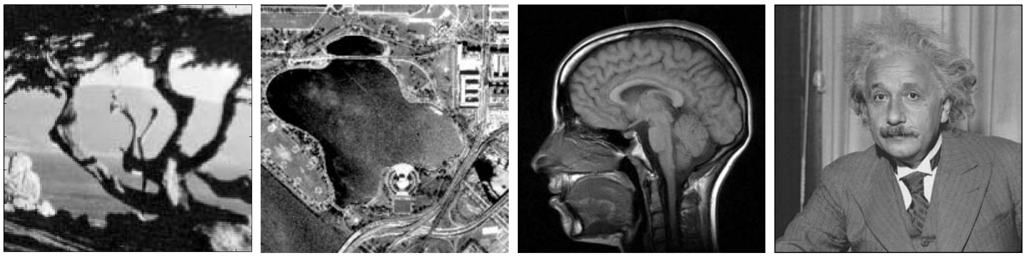

Figure 4.10: Tree, DC, and MRI, and Einstein images

To give an example of the performance improvement offered by such a filter design methodology, we compare the bias magnitude for the range of translations $v_{0_1}, v_{0_2} \in [-2, 2]$ for several popular filter sets. We measure overall performance by averaging the magnitude of

	Tree	DC Sat.	MRI	Einstein
Central	0.098	0.137	0.124	0.107
Fleet	0.130	0.183	0.162	0.150
Nestares	0.061	0.086	0.080	0.063
Elad	0.112	0.074	0.056	0.063
Optimized	0.048	0.064	0.061	0.045

Figure 4.11: Overall registration error $\overline{Err}$ for the range $v_{0_1}, v_{0_2} \in [-2, 2]$

the registration RMSE for the different filter sets over a set of translations in this test range. The performance measure is given by

$$\overline{Err} = \frac{1}{N_S} \sum_{\mathbf{v}_0 \in S_{\mathbf{v}}} \overline{rmse}(\mathbf{v}_0) \tag{4.40}$$

where $S_{\mathbf{v}}$ is the set of test translation points of size N_S. We choose this performance measure as it shows overall performance error in units of pixels. If $SNR = \infty$ (no noise added to the pair of images), then $\overline{rmse}(\mathbf{v}_0) = \frac{1}{2}\|\mathbf{b}(\mathbf{v}_0)\|$. In a sense, the overall error is a measure of the average magnitude error over a range of translations. For our experiments, we assume that the range of translations is uniformly sampled in the test range.

We first examine the zero-noise case where $SNR = \infty$. Such a scenario corresponds to the typical experimental setup examined in gradient-based estimation literature where rarely is noise added to the images prior to estimation [14, 61]. Under such conditions, only the deterministic estimator bias affects the overall estimator performance. For our experiment, the we uniformly sampled the region $[-2, 2] \times [-2, 2]$ in increments of $[\frac{1}{10}, \frac{1}{10}]$ pixels to generate our test set $S_{\mathbf{v}}$ of translations. The filters compared were the simple central difference filter (Central), the 2nd order derivative filter mentioned in [14] (Fleet), the Nestares filter and the set of filters designed using the method of [61] (Elad). All of the filters have 5 taps (2 coefficients) except the filters of [61] which were 9 tap filters. Prior to estimation, the images were pre-filtered either with a sampled Gaussian pre-smoothing filter with standard deviation of $\sqrt{(3)}$ pixels or the specially tuned filters of Elad. The performance for each filter set are shown in the table in Figure 4.11 for the images in Figure 3.2. The optimized filter shows improved overall performance for all images except for the MRI image, where the optimized filter performance was only slightly worse than that of Elad. Recalling that the Elad filters were 9-tap filters as opposed to the 5-tap optimized filters we see that, in general, the proposed filters improve average estimator performance while realizing computational savings. Furthermore, we found that when using larger optimized filters, we can achieve even greater improvement over the other filters. This improved performance results from the increases degrees of freedom of the optimization routine. Basically, larger filters allow for more precision in tuning the frequency response of the filters. We shall show this momentarily.

To visualize the effect of the optimized filters, Figure 4.12 shows the bias magnitude $\|\mathbf{b}(\mathbf{v}_0)\|$ in registering the Tree. The top graph shows the bias magnitude when the [1] (Nestares) filter was used (the second best filter). The bottom graph shows the bias magnitude when using the filters designed by optimizing (4.39).

From the bias exhibited in Figure 4.12, we see that the bias magnitude primarily depends on the magnitude of the translation $\|\mathbf{v}_0\|$. Furthermore, Figure 4.12 reveals the polar symmetry of the registration bias. Because of this symmetry, we plot the magnitude of registration bias for the collection of filter sets for the set of translations $v_{0_1} = v_{0_2} \in [0,\ 2]$ in Figure 4.13. This representative slice reveals the important performance characteristics of each filter set. Figure 4.13 compares the bias magnitude for all of the filters when registering the DC Satellite image. Here, we see that while the bias of all the filters becomes severe as the magnitude of the translation increases, the bias for the optimizing filter is minimized. The optimized filters have the coefficients $g_1 = [0.8792,\ -1.2459,\ 0\ 1.2459,\ -0.8792]$ and $g_2 = [0.8969,\ -1.2606,\ 0\ 1.2606,\ -0.8969]^T$.

To evaluate the performance of the optimized filter in a more realistic scenario, we must compare estimator performance in the presence of noise. To this end, we conduct Monte Carlo (MC) simulations at SNR ranging from about 10 dB through 60 dB.[2] At each SNR, we measure the MSE in estimating $\mathbf{v}_0$ along the line $v_{0_1} = v_{0_2} \in [0, 2]$ in increments of $\frac{1}{10}$

[2] The SNR is measured as $SNR = 10 \log_{10} \frac{\sigma_f^2}{\sigma^2}$ where σ_f^2 is the variance of the clean frame and σ^2 the variance of the noise.

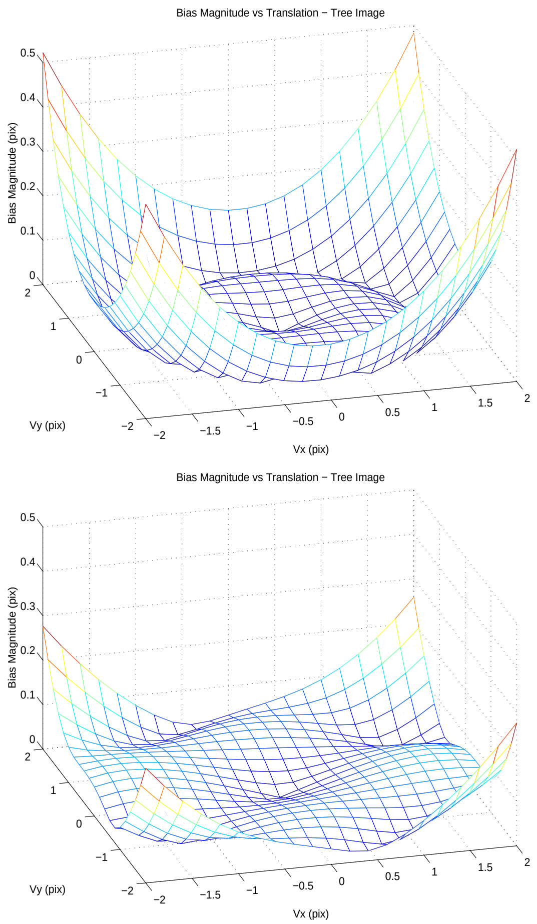

Figure 4.12: Magnitude of estimator bias $\|\mathbf{b}(\mathbf{v}_0)\|$ vs translation using the Nestares gradient filters (top) and the bias minimizing gradient filters (bottom).

pixels by averaging the square estimator error over 1000 MC runs. As before, we use the same experimental setup used to produce Figure 4.13 in terms of filter sets. Here, we see that the optimized filters continue to outperform the other filters over the wide range of SNR. We note that the performance does not vary widely until very low SNR (12 dB) as the bias dominates the MSE as shown in [76]. Essentially, Figure 4.14 shows that the optimized filters retain their competitive performance over a wide range of imaging SNR for non-iterative registration.

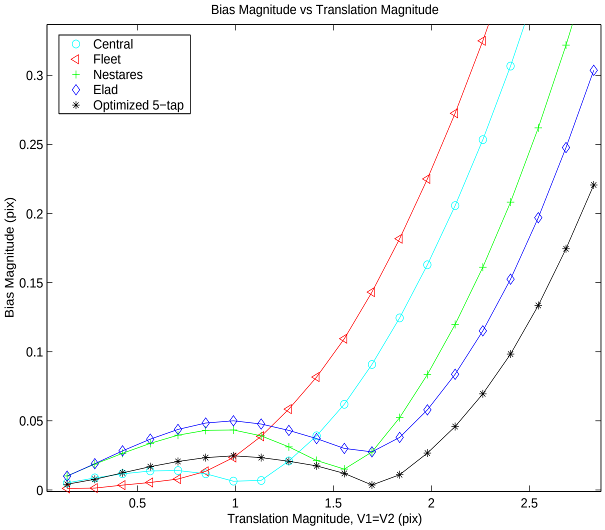

Figure 4.13: Magnitude of estimator bias $\|\mathbf{b}(\mathbf{v}_0)\|$ vs translation magnitude $\|\mathbf{v}_0\|$ where $v_{0_1} = v_{0_2}$

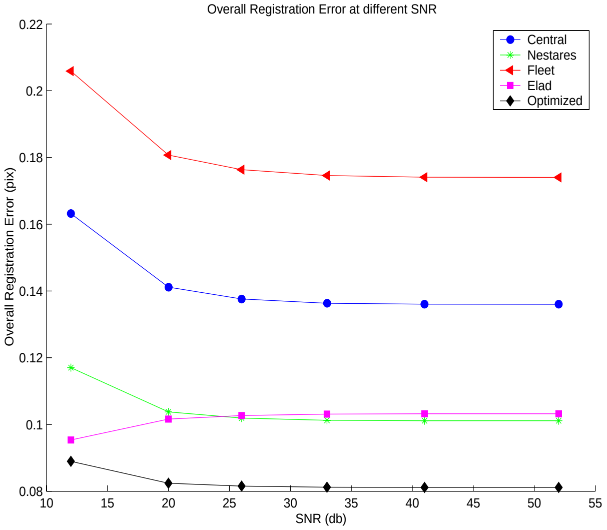

Figure 4.14: Overall estimation error $\overline{Err}$ at different SNR over $v_{0_1} = v_{0_2} \in [0, 2]$ for the Tree image.

Filter Design for Multiscale Iterative Registration

Traditionally, the same gradient filter has been applied at each level of the pyramid during multiscale gradient-based estimation [36]. The performance and rate of convergence of the multiscale method can be further improved using optimally designed bias-minimizing filters. We suggest the novel approach of designing *different* gradient filters for each level of

	Tree	DC Sat.	MRI	Einstein
Central	0.006	0.010	0.004	0.012
Fleet	8.14e-4	0.002	0.001	0.008
Nestares	0.012	0.018	0.011	0.020
Elad	0.010	0.006	0.001	0.015
Optimized	2.07e-4	5.57e-4	2.57e-4	0.006

Figure 4.15: Overall registration error $\overline{Err}$ for multiscale estimation over the range $v_{0_1}, v_{0_2} \in [-6, 6]$

the pyramid, each according to the cost function (4.39). Optimizing gradient filters in such a manner improves the convergence rates of the iterative estimation by reducing the residual motion left over from biased estimates produced from earlier iterations. More importantly, minimizing the estimator reduces the possibility of the iterative estimation process diverging, thereby offering a more stable method of estimation. Furthermore, since at every iteration the residual motion to be estimated is reduced, we propose designing filters which assume that the ranges of translation shrink as the iterations proceed.

To show an example of such optimized filters for the multiscale registration scenario, we design gradient filters for a three level multiscale pyramid. As in Section 4.4.2, we first examine the zero-noise scenario ($SNR = \infty$) where only the bias contributes to estimator MSE. The optimized gradient filters were designed for the translation ranges $v_{0_1}, v_{0_2} \in [-2, 2], [-.5, .5], [-.2, .2]$ for each of the three pyramid levels. Figure 4.15 shows the overall multiscale registration error over the translation test set $v_{0_1}, v_{0_2} \in [-6, 6]$ uniformly sampled with a spacing of $[\frac{1}{5}, \frac{1}{5}]$ pixels. Again, we see that the optimized filters offer superior performance for multiscale estimation in terms of the registration error over a wide range of translations.

As before, to visualize the estimator performance in the multiscale setting, the registration error for the Tree image is plotted in Figure 4.16 along the line $v_{0_1} = v_{0_2} \in [0, 6]$ for the zero-noise scenario. While all of the estimators show significant improvement over the non-multiscale iterative approach, the bias-minimizing 5-tap filters offer consistent improvement in estimator accuracy over the entire range of translations. For practical applications, the registra-

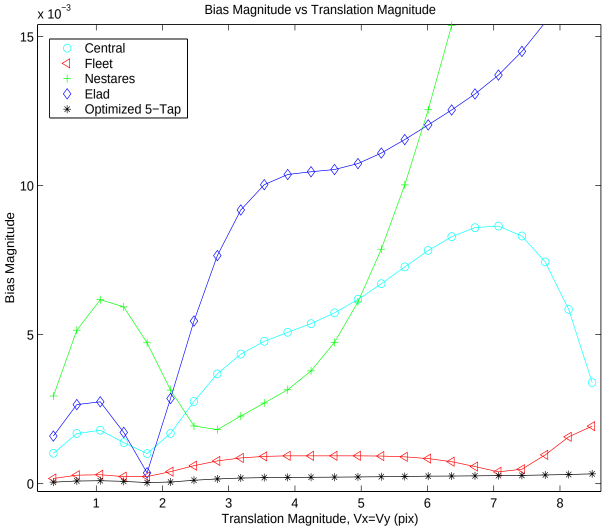

Figure 4.16: Magnitude of registration bias $\|\mathbf{b}(\mathbf{v})\|$ vs translation magnitude $\|\mathbf{v}\|$ where $v_{0_1} = v_{0_2}$ for the Tree image.

tion error is so small as to be considered almost unbiased. Overall, we see that principled filter design offers improvement for multiscale image registration.

Again, we evaluate the performance of the optimized filters at different imaging SNR. As before, we perform MC simulations at each SNR to measure the MSE of the multiscale approach in estimating $\mathbf{v}$ along the line $v_{0_1} = v_{0_2} \in [0, 6]$ this time in increments of $\frac{1}{2}$. As before, we use the same experimental setup used to produce Figure 4.14, only this time we use the multiscale approach. Here, we see that the optimized filters outperform the other filters for SNR greater than about 25 dB. In fact, below this SNR, the performance of the optimized filters for multiscale estimation degrade substantially. In this SNR regime, the MSE is no longer dominated by estimator bias. It is apparent that at this SNR, minimizing bias is no longer a suitable objective for improving overall performance in the sense of MSE. We note, however, that SNR below 25 dB represents a very noisy scenario not often encountered in typical video imaging and rarely, if ever, addressed in the gradient-based motion estimation literature.

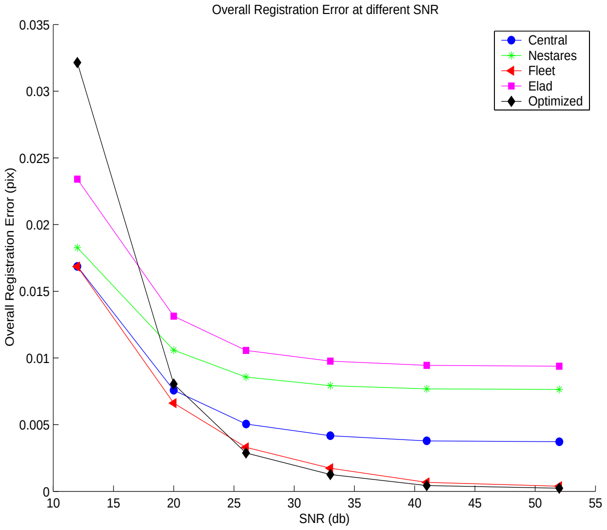

Figure 4.17: Overall multiscale estimation error $\overline{Err}$ at different SNR over $v_{0_1} = v_{0_2} \in [0, 6]$ for the Tree image.

4.5 Conclusion

In this chapter we have presented detailed analysis of the bias associated with gradient-based algorithms. Detailed analysis of the estimator bias gives the practitioner of motion estimation a keen insight into the performance tradeoffs associated with gradient-based algorithms. In particular, our analysis explores the bias as it relates to image content, motion magnitude, and choice of gradient-filters. Using this bias expression we have been able to present a general rule-of-thumb MSE performance bound using the complete CR bound for gradient-based estimation at high SNR. We believe such information will prove critical to the design and understanding of systems relying on the output of gradient-based algorithms. Lastly, we have presented a novel approach to designing gradient filters for the gradient-based algorithm which reduce estimator bias and have verified the bias minimizing properties of such filters. We believe that the next step in filter design will address a MSE minimizing criterion.

4.A 2-D Bias and CR Bound for Gradient-Based Estimation

In this appendix, we derive the bias equations for the 2-dimensional case similar to those in Chapter 4 and incorporate this bias function into the complete CR bound of (3.1). Here we use vector notation. Namely, $\mathbf{v}_0 = [v_{0_1},\, v_{0_2}]^T$ and $\underline{\theta} = [\theta_1,\, \theta_2]^T$ and $\mathbf{k} = [n_1,\, n_1]^T$. Thus, we write the data model as

$$z(\mathbf{k}) = f(\mathbf{k} + \mathbf{v}_0) - f(\mathbf{k}) + \epsilon(\mathbf{k}) \tag{4.41}$$

We proceed to derive the bias directly in the frequency domain. The shifted sequence $f(\mathbf{k}+\mathbf{v}_0)$ has a DTFT of $F(\underline{\theta})e^{j(\underline{\theta}^T\mathbf{v}_0)}$ and the DTFT of the data model becomes

$$Z(\underline{\theta}) = F(\underline{\theta})\left[e^{j(\underline{\theta}^T\mathbf{v}_0)} - 1\right] + \xi(\underline{\theta}). \tag{4.42}$$

We expand the exponential in a Taylor series $e^{j(\underline{\theta}^T\mathbf{v}_0)} = 1 + j(\underline{\theta}^T\mathbf{v}_0) - ...$ and truncate after the linear term to obtain the formula $Z(\underline{\theta}) \approx jF(\underline{\theta})\ \underline{\theta}^T\mathbf{v}_0 + \xi(\underline{\theta})$ from which we obtain the linear estimator

$$\hat{\mathbf{v}}_0 = \mathbf{F}^{-1}\int |F(\underline{\theta})|^2 j\underline{\theta} Z^*(\underline{\theta})d\underline{\theta} \tag{4.43}$$

where $\mathbf{F} = \int |F(\underline{\theta})|^2\left[\underline{\theta\theta}^T\right] d\underline{\theta}$.

Similar to the 1-D case, the expected value of the estimate is

$$E[\hat{\mathbf{v}}_0] = \mathbf{F}^{-1}\int |F(\underline{\theta})|^2\underline{\theta}\sin(\underline{\theta}^T\mathbf{v}_0)d\underline{\theta}. \tag{4.44}$$

To obtain this form, we have made the same simplification as in Section 3 wherein the imaginary portion of the integrand is removed as it is an odd function, hence integrating to zero. Thus, we obtain the bias function

$$\mathbf{b}(\mathbf{v}_0) = \mathbf{F}^{-1}\int |F(\underline{\theta})|^2\underline{\theta}\sin(\underline{\theta}^T\mathbf{v}_0)d\underline{\theta} - \mathbf{v}_0 \tag{4.45}$$

To analyze this bias function, we approximate the sinusoid function within the integrand as a truncated Taylor series expansion about $v = 0$ as $\sin(\underline{\theta}^T\mathbf{v}_0) \approx \underline{\theta}^T\mathbf{v}_0 - \frac{1}{6}(\underline{\theta}^T\mathbf{v}_0)^3$. Noting that

$\underline{\theta}^T \mathbf{v}_0 = |\mathbf{v}_0| \underline{\theta}^T \mathbf{n}_\psi$ where $\mathbf{n}_\psi$ is the unit vector $[\cos(\psi),\, \sin(\psi)]^T$ we can approximate the bias function as

$$\begin{aligned} \mathbf{b}(\mathbf{v}_0) &\approx \mathbf{F}^{-1} \int |F(\underline{\theta})|^2 \underline{\theta} \left[\underline{\theta}^T \mathbf{v}_0 - (\underline{\theta}^T \mathbf{v}_0)^3 \right] d\underline{\theta} - \mathbf{v}_0 \\ &= \mathbf{v}_0 - \frac{1}{6} \mathbf{F}^{-1} \int |F(\underline{\theta})|^2 \underline{\theta} (\underline{\theta}^T \mathbf{v}_0)^3 d\underline{\theta} - \mathbf{v}_0 \\ &= -\frac{|\mathbf{v}_0|^3}{6} \mathbf{F}^{-1} \int |F(\underline{\theta})|^2 \underline{\theta} (\underline{\theta}^T \mathbf{n}_\psi)^3 d\underline{\theta} = -\frac{|\mathbf{v}_0|^3}{6} \mathbf{F}^{-1} \mathbf{d} \end{aligned} \tag{4.46}$$

where $\mathbf{d} = \int |F(\underline{\theta})|^2 \underline{\theta} (\underline{\theta}^T \mathbf{n}_\psi)^3 d\underline{\theta}$. Thus, the bias behaves as a cubic function of the translation magnitude $|\mathbf{v}_0|$ where the coefficient depends on the spectrum of the image.

As with the 1-D case, in practice we must approximate the gradients using gradient kernels $g_1(\mathbf{k})$ and $g_2(\mathbf{k})$ which have corresponding frequency representations $G_1(\underline{\theta})$ and $G_2(\underline{\theta})$ or in vector notation $\mathbf{G}(\underline{\theta}) = [G_1(\underline{\theta}),\, G_2(\underline{\theta})]^T$. This produces the estimator,

$$\hat{\mathbf{v}}_0 = \mathbf{F}^{-1} \int |F(\underline{\theta})|^2 j \mathbf{G}(\underline{\theta}) Z^*(\underline{\theta}) d\underline{\theta} \tag{4.47}$$

where now $\mathbf{F} = \int |F(\underline{\theta})|^2 \left[\mathbf{G}(\underline{\theta}) \mathbf{G}(\underline{\theta})^T \right] d\underline{\theta}$. Using the same low-noise assumptions that we made in Section 3, we examine only the deterministic bias which is

$$\mathbf{b}(\mathbf{v}_0) = \mathbf{F}^{-1} \int |F(\underline{\theta})|^2 \mathbf{G}(\underline{\theta}) \sin(\underline{\theta}^T \mathbf{v}_0) d\underline{\theta} - \mathbf{v}_0 \tag{4.48}$$

Using these equations for the bias, we can now derive the full CRLB for gradient-based estimation of 2-D translation. We first note that

$$\frac{\partial E[\hat{\mathbf{v}}_0]}{\partial \mathbf{v}_0} = \mathbf{F}^{-1} \int |F(\underline{\theta})|^2 \left[\mathbf{G}(\underline{\theta}) \underline{\theta}^T \right] \cos(\underline{\theta}^T \mathbf{v}_0) d\underline{\theta} = \mathbf{Z}. \tag{4.49}$$

Using this equation, we obtain for the complete CR bound for the 2-D case as

$$MSE(\mathbf{v}_0) \geq \mathbf{Z} \mathbf{J}^{-1} \mathbf{Z}^T + \mathbf{b}(\mathbf{v}_0) \mathbf{v}(\mathbf{v}_0)^T \tag{4.50}$$

where $\mathbf{J}$ represents the Fisher Information matrix derived in Chapter 3.

4.B Projection-Based vs Direct Gradient-Based Estimator Bias

Armed with the bias expressions for the gradient-based algorithm, we can begin to understand the performance improvement from the use of projections in estimating translation reported in Chapter 2. We now offer some intuition behind the improved performance of the gradient-based algorithm using projections. Basically, we compare the bias associated with the projection-based algorithm, denoted $\mathbf{b}_p(\mathbf{v}_0)$ with the bias associated with the direct estimation which we denote as $\mathbf{b}_d(\mathbf{v}_0)$.

From the well known Projection-Slice Theorem, we see that the Fourier transform of the projection of the image $f(x_1, x_2)$ at an angle ϕ is equivalent to a slice of the Fourier transform of the 2-D image function $F(\theta_1, \theta_2)$ through the origin at an angle ϕ [23]. Using this theorem, we can relate the bias of the 1-D projection-based algorithm to that of the 2-D direct algorithm.

We now address the simple case where only a pair of projections at $\phi = 0$ and $\phi = \frac{\pi}{2}$ are used in the projection-based algorithm. Thus, the corresponding Fourier transforms of the projected image function $r(p, \phi)$ at these angles is given by

$$\begin{aligned} \mathcal{F}\left[r(p,0)\right] &= F(\theta_1, 0) \\ \mathcal{F}\left[r\left(p, \frac{\pi}{2}\right)\right] &= F(0, \theta_2) \end{aligned} \tag{4.51}$$

Using (4.9), we see that the components of the bias $\mathbf{b}_p(\mathbf{v}_0)$ for the projection-based estimation are given by

$$[\mathbf{b}_p(\mathbf{v}_0)]_1 = \frac{\int_{-\pi}^{\pi} |F(\theta_1, 0)|^2 G_1(\theta_1) \sin(v_{0_1}\theta_1) d\theta_1}{\int_{-\pi}^{\pi} |F(\theta_1, 0)|^2 G_1^2(\theta_1) d\theta_1} - v_{0_1} \tag{4.52}$$

$$[\mathbf{b}_p(\mathbf{v}_0)]_2 = \frac{\int_{-\pi}^{\pi} |F(0, \theta_2)|^2 G_2(\theta_2) \sin(v_{0_2}\theta_2) d\theta_2}{\int_{-\pi}^{\pi} |F(0, \theta_2)|^2 G_2^2(\theta_2) d\theta_2} - v_{0_2} \tag{4.53}$$

We use the subscript p to indicate the 1-D or projection based estimator bias. Here we see that the components of the bias function depend only on their respective translational component. This separability does not apply to the bias associated with the direct 2-D estimation algorithm.

To simplify the presentation we make the assumption that the image spectrum has the following symmetry

$$
\begin{aligned}
|F(\theta_1, \theta_2)| &= |F(-\theta_1, \theta_2)| && (4.54)\\
|F(\theta_1, \theta_2)| &= |F(\theta_1, -\theta_2)| && (4.55)
\end{aligned}
$$

We make this assumption to simplify the matrix **F** defined by in Appendix 4.A. With such symmetry, the matrix **F** is given by

$$
\begin{aligned}
[\mathbf{F}]_{11} &= \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |F(\theta_1, \theta_2)|^2 G_1^2(\theta_1) d\theta_1 d\theta_2\\
[\mathbf{F}]_{12} &= [\mathbf{F}]_{21}\\
&= \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |F(\theta_1, \theta_2)|^2 G_1(\theta_1) G_2(\theta_2) d\theta_1 d\theta_2\\
&= 0\\
[\mathbf{F}]_{22} &= \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |F(\theta_1, \theta_2)|^2 G_2^2(\theta_2) d\theta_1 d\theta_2
\end{aligned}
$$

In other words, because of the symmetry assumption, the off-diagonal terms of **F** are zero. In practice, natural images whose magnitude spectra approximately follow $|F(\theta_1, \theta_2)| \approx \frac{1}{\sqrt{\theta_1^2+\theta_2^2}}$ posses such symmetry.

With the simplified form for the matrix **F**, the components of the estimator bias for the direct 2-D algorithm $\mathbf{b}_d(\mathbf{v}_0)$ can be expressed as

$$
\begin{aligned}
[\mathbf{b}_d(\mathbf{v}_0)]_1 &= \frac{\int\int |F(\theta_1, \theta_2)|^2 G_1(\theta_1) \sin(v_{0_1}\theta_1 + v_{0_2}\theta_2) d\theta_1 d\theta_2}{\int\int |F(\theta_1, \theta_2)|^2 G_1^2(\theta_1) d\theta_1 d\theta_2} - v_{0_1}\\
&= \frac{\int\int |F(\theta_1, \theta_2)|^2 G_1(\theta_1) \sin(v_{0_1}\theta_1) \cos(v_{0_2}\theta_2) d\theta_1 d\theta_2}{\int\int |F(\theta_1, \theta_2)|^2 G_1^2(\theta_1) d\theta_1 d\theta_2} - v_{0_1} && (4.56)\\
[\mathbf{b}_d(\mathbf{v}_0)]_2 &= \frac{\int\int |F(\theta_1, \theta_2)|^2 G_2(\theta_2) \sin(v_{0_1}\theta_1 + v_{0_2}\theta_2) d\theta_1 d\theta_2}{\int\int |F(\theta_1, \theta_2)|^2 G_2^2(\theta_2) v d\theta_1 d\theta_2} - v_{0_2}\\
&= \frac{\int\int |F(\theta_1, \theta_2)|^2 G_2(\theta_2) \cos(v_{0_1}\theta_1) \sin(v_{0_2}\theta_2) d\theta_1 d\theta_2}{\int\int |F(\theta_1, \theta_2)|^2 G_2^2(\theta_2) d\theta_1 d\theta_2} - v_{0_2} && (4.57)
\end{aligned}
$$

Again, the subscript 2 indicates the direct or 2-D algorithm. The simplified form of (4.56) and (4.57) results from the symmetry assumptions of (4.54) and (4.55).

From (4.56) and (4.57), we see that the each component of the bias vector depends on both components of the translation vector $\mathbf{v}_0$. For example, the bias associated with estimating v_{0_1} also depends on the translation parameter v_{0_2} by way of the $\cos$ term in the numerator of (4.56). For example, Figure 4.18 shows the bias magnitude surface $\|\mathbf{b}(\mathbf{v}_0)\|$ for both the projection-based as well as the direct gradient-based estimation algorithms for the Tree image. To generate these surface plots, no pre-smoothing filters were applied to the images. We see

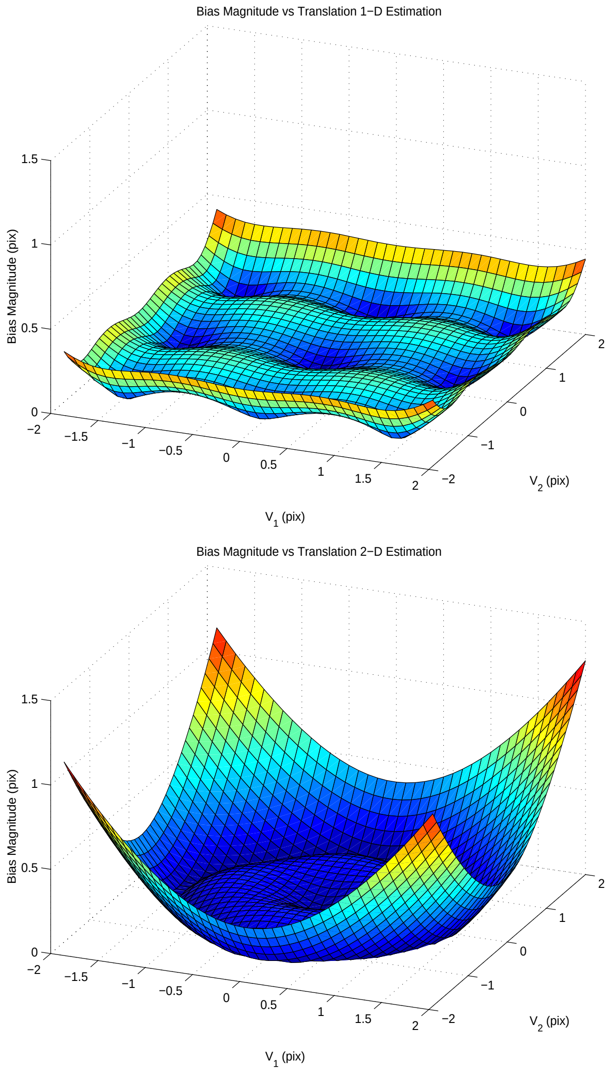

Figure 4.18: Magnitude of estimator bias $\|\mathbf{b}(\mathbf{v}_0)\|$ vs translation using the Nestares gradient filters [1] for the projection-based algorithm (top) and the direct algorithm (bottom).

in Figure 4.18 that the difference in bias between the 1-D and 2-D algorithms is most severe for large translation magnitudes $\|\mathbf{v}_0\|$. We observe this behavior for a wide variety of images. We have already shown that both the 2-D and 1-D bias expressions behave approximately as a cubic function of $\|\mathbf{v}_0\|$ and $|v_0|$ respectively. Because of the separability of the projection-based algorithm, the 2-D bias of the projection-based algorithm grows approximately as a cubic function of $\max(|v_{0_1}|, |v_{0_2}|)$. The bias of the direct 2-D algorithm, however, grows as a cubic function of $\|\mathbf{v}_0\|$. Thus, for large translation magnitudes $\|\mathbf{v}_0\|$, the bias of the projection-based is less than that of the direct algorithms. We note that the average bias magnitude $\|\mathbf{b}(\mathbf{v}_0)\|$ for the surface plots shown in Figure 4.18 are 0.164 for the projection-based approach and 0.268 for the direct approach. In other words, on average, the bias of the projection-based approach has significantly magnitude over the range $[-2, 2] \times [-2, 2]$ than the direct approach. This property of the projection-based estimator explains, in part, the improvement in performance of the projection-based algorithm over the direct for translation estimation.

Chapter 5

Performance Analysis of Multiframe Registration of Aliased Images

5.1 Introduction

In Chapter 3, we studied the performance bound for pair-wise image registration assuming that the images were sampled above the Nyquist rate. In this chapter, we extend these results to the scenario where the images are sampled below the Nyquist rate and hence contain aliased information. Furthermore, we show that a natural consequence of aliased imaging is the need for multiframe registration. We will show that this estimation problem is intimately related to the problem of super-resolution. In general, the problem of super-resolution can be expressed as that of combining a set of noisy, low-resolution, blurry images to produce a higher resolution image or image sequence. In the last decade, several papers have proposed algorithms addressing the problem of super-resolution. [63] offers a broad review of the work this area. With some simplifying assumptions, the estimation problem is typically divided into the tasks of first registering the low resolution images with respect to the coordinate system of the desired high resolution image followed by fusing the low-resolution data (reconstruction) and finally deblurring and interpolating to produce the final high resolution image (restoration).

Historically, most research in super-resolution has tended to focus on the latter stages assuming that generic image registration algorithms could be trusted to produce estimates with a high level of accuracy. As we shall show, however, such an approach is necessarily sub-optimal at best or inherently biased. Relatively recently, researchers have noted the importance of solving the estimation problems of image registration and super-resolution in a joint fashion [64–66]. Conversely, the only paper (to our knowledge) concerning registration of aliased (sub-Nyquist) images [43], does not directly address the problem of image restoration during registration. Instead, it focuses on mitigating generic (none image specific) effects of aliasing on the registration algorithm. The one other paper which claims to addresses sub-pixel translation estimation between a pair of downsampled images, makes the assumption that ”no spectral fold-over (overlap) occurs” after downsampling [70]. In other words, the images are downsampled, but contain no aliased information. We shall show that the performance analysis (and algorithmic design) for the sub-Nyquist scenario must study the problems of image registration and image reconstruction in a joint fashion.

Similar to Chapter 3, we analyze the joint problem of image registration and its related counterpart (high resolution image reconstruction) in the context of the Cramér-Rao inequality. To date, no work has addressed the performance limits associated with the registration of aliased images. In this chapter, we primarily study the MSE performance bound on sub-Nyquist image registration. We also address the problem of image reconstruction as it is a natural byproduct of proper image registration. Finally, we outline the relationship of the sub-Nyquist image registration problem to the problem of super-resolution.

This chapter is organized as follows. In Section 5.2, we derive the Fisher Information Matrix (FIM) for the joint problem of image registration and reconstruction. With the Fisher Information, we present the Cramér-Rao (CR) inequality bounding the MSE for the class of unbiased estimators. In Section 5.3, we analyze the performance bounds and offer insight into the inherent challenges and tradeoffs in the registration of aliased images. In Section 5.4, we study the influence of prior information on the joint problem of sub-Nyquist image registration.

Finally, in Section 5.6, we summarize the contribution of this work and suggest future research directions.

5.2 CR Bound on the Registration of Aliased Images

For the general problem of registering aliased images, it is assumed that we are given a set of low resolution images which consist of noisy, warped, blurred, and downsampled versions of an unknown high resolution image. As we studied in Chapter 3, we focus on the motion captured by a global shift or a translation between frames.

To simplify the notation and remain consistent with the related problem of super-resolution, we formulate the data model using matrix notation making a notational departure from the model of 1.1. To do so, we represent the samples of image function $f(x_1, x_2)$ at the sample locations in vector form by raster scanning the samples as

$$\mathbf{f} = \begin{pmatrix} f(0,0) \\ \vdots \\ f(N_H, 0) \\ f(0,1) \\ \vdots \\ f(N_H, N_H) \end{pmatrix} \tag{5.1}$$

Using a similar raster scanning procedure we use $\mathbf{z}_k$ to represent the measured image at the sampled time $t = kT_t$. For such an assumption, we represent the forward process by the linear equation

$$\mathbf{z}_k \quad = \quad \mathbf{D}\mathbf{U}(\mathbf{v}_k)\mathbf{f} + \mathbf{e}_k, \;\; k = 0 \ldots K \tag{5.2}$$

The vectors $\mathbf{z}_k$ represent the samples of the measured images scanned in some fashion to form N_L dimensional vectors. Likewise, $\mathbf{f}$ represents the unknown original high resolution image similarly scanned to form a N_H dimensional vector. The matrix $\mathbf{D}$ captures the downsampling operation (which leads to aliased images), and $\mathbf{U}(\mathbf{v}_k)$ the translational motion operation with $\mathbf{v}_k = [v_{k_1}, v_{k_2}]^T$ being the unknown translation vector for measured frame k. Finally, $\mathbf{e}_k$ represents the vector of additive white Gaussian measurement noise with variance σ^2.

For the purpose of this chapter, we make several additional assumptions about our forward model (5.2). First, we assume that the unknown high resolution image $\mathbf{f}$ is a bandlimited image sampled above the Nyquist rate. In other words, the unknown high resolution image does not contain aliasing, but the noisy measurement images do contain aliasing. From this assumption, the matrix $\mathbf{U}(\mathbf{v}_k)$ (which we will refer to as $\mathbf{U}_k$) representing the translational shift of the image $f(n_1 - v_{0_1}, n_2 - v_{0_2})$, reflects a convolution operation with a shifted 2 dimensional sinc function. In other words,

$$f(n_1 - v_{0_1}, n_2 - v_{0_2}) = f(n_1, n_2) * *\text{sinc}(n_1 - v_{0_1}, n_2 - v_{0_2})$$

Such a motion formulation allows arbitrary, possibly non-integer shifts. The matrix $\mathbf{U}_k$ has the property that $\mathbf{U}_k^T \mathbf{U}_k = \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. In other words, shifting the image followed by a shift in the reverse direction does not change the pixel values of the high resolution image. Furthermore, we note that when the motion vector $\mathbf{v}_k$ reflects integer shifts (in units of high resolution pixels), then the matrix $\mathbf{U}_k$ is simply a permutation of the identity matrix $\mathbf{I}$. Second, we assume that the downsampling operation is based on a known downsampling factor M where $\frac{N_L}{N_H} = \frac{1}{M^2}$. For our purposes, we assume that the downsampling factor M is an integer. The M^2 come from the assumption that the downsampling factor for both the x_1 and x_2 dimensions is M. Thus, $\mathbf{D}$ is an N_L by N_H matrix representing the downsampling operation. Third, in our formulation, we suppose that $K + 1$ low resolution measured images are available. Without loss of generality, we assume that the initial image $\mathbf{z}_0$ dictates the coordinate system so that $\mathbf{U}_0 = \mathbf{I}$ and hence we only have to estimate K unknown translation vectors $\mathbf{v}_k$ during the super-resolution process for a given set of $K + 1$ low resolution frames.

Depending on the application, it is natural to distinguish the problem of estimating the translational motion parameters $\{\mathbf{v}_k\}$ from the estimation of the image $\mathbf{f}$. To simplify the notation, we define $\vec{\text{v}}$ to be the set of all unknown motion parameters, or $\vec{\text{v}} = [v_{1_1}, v_{1_2}, \ldots, v_{K_1}, v_{K_2}]^T$. Because of this dichotomy, we show the Fisher Information Matrix $\mathbf{J}(\mathbf{f}, \vec{\text{v}})$ using the following partitioned structure

$$\mathbf{J}(\mathbf{f}, \vec{\text{v}}) = \begin{pmatrix} \mathbf{J}_{\mathbf{ff}} & \mathbf{J}_{\mathbf{f}\vec{\text{v}}} \\ \mathbf{J}_{\mathbf{f}\vec{\text{v}}}^T & \mathbf{J}_{\vec{\text{v}}\vec{\text{v}}} \end{pmatrix} \tag{5.3}$$

where the matrix $\mathbf{J}_{\mathbf{ff}}$ captures the available information solely pertaining to the unknown image $\mathbf{f}$, and $\mathbf{J}_{\vec{\text{v}}\vec{\text{v}}}$ the information pertaining the motion parameters $\vec{\text{v}}$, and $\mathbf{J}_{\mathbf{f}\vec{\text{v}}}$ reflects the *information inter-correlation* between the two sets of unknown parameters. Were $\mathbf{J}_{\mathbf{f}\vec{\text{v}}} = 0$, then the problem of image reconstruction could be de-coupled from the problem of sub-Nyquist image registration. As we shall show momentarily, such structure is impossible except for degenerative cases which are of no practical interest. Thus, we argue that the problems of sub-Nyquist registration and image reconstruction must be solved in a joint fashion. Consequently, throughout this chapter we study the performance bound on image reconstruction as a byproduct of our analysis.

Although the estimation must be performed jointly, based on the block decomposition of (5.3), we can separate the performance analysis for the two estimation problems using the block matrix inversion principle [67]. Using this principle, the inverted FIM (and hence the CR bound) is given by

$$\mathbf{J}^{-1}(\mathbf{f}, \vec{\text{v}}) = \begin{pmatrix} \mathcal{S}_{\mathbf{f}}^{-1} & \mathbf{J}_{\mathbf{ff}}^{-1}\mathbf{J}_{\mathbf{f}\vec{\text{v}}}\mathcal{S}_{\vec{\text{v}}}^{-1} \\ \mathcal{S}_{\vec{\text{v}}}^{-1}\mathbf{J}_{\mathbf{f}\vec{\text{v}}}^T\mathbf{J}_{\mathbf{ff}}^{-1} & \mathcal{S}_{\vec{\text{v}}}^{-1} \end{pmatrix} \tag{5.4}$$

where the $\mathcal{S}$ matrices are the Schur matrix complements given by

$$\mathcal{S}_{\vec{\text{v}}} = \mathbf{J}_{\vec{\text{v}}\vec{\text{v}}} - \mathbf{J}_{\mathbf{f}\vec{\text{v}}}^T\mathbf{J}_{\mathbf{ff}}^{-1}\mathbf{J}_{\mathbf{f}\vec{\text{v}}} \tag{5.5}$$

$$\mathcal{S}_{\mathbf{f}} = \mathbf{J}_{\mathbf{ff}} - \mathbf{J}_{\mathbf{f}\vec{\text{v}}}\mathbf{J}_{\vec{\text{v}}\vec{\text{v}}}^{-1}\mathbf{J}_{\mathbf{f}\vec{\text{v}}}^T \tag{5.6}$$

In this block partitioned formulation, we see a certain symmetry of the two estimation problems. At first glance, we observe that there is a net loss in information due to the interdependence of the two estimation problems because the second terms in (5.5) and (5.6) are positive semidefinite matrices. For instance, in the case of translation estimation, the term $\mathbf{J}_{\mathbf{f}\vec{v}}^{T}\mathbf{J}_{\mathbf{ff}}^{-1}\mathbf{J}_{\mathbf{f}\vec{v}}$ represents the orthogonal projection of the information about the registration parameters projected onto the linear subspace encompassing the information about the unknown image $\widetilde{\mathbf{f}}$ [68]. The Fisher Information captures the relationship between small perturbations of the unknown signal parameters and the likelihood function of the measured data. As such, the net loss of information reflects the ambiguity arising from a small perturbation in either sets of signal parameters producing the same perturbation in the likelihood function. Simply put, such a structure captures the ability to distinguish variations in the measured data as depending on one parameter set versus the other.

Typically, for the problem of image reconstruction, we are interested in a performance measure reflecting the goal of reconstructing an entire image $\mathbf{f}$. One natural performance metric is the component-wise MSE summed over all pixels in the image. The CR bound for such a measure of image restoration over the entire image is given by

$$T(\mathbf{f}) = \left(\frac{tr(MSE(\mathbf{f}))}{N_H}\right)^{\frac{1}{2}} \quad \geq \quad \left(\frac{tr(\mathcal{S}_{\mathbf{f}}^{-1})}{N_H}\right)^{\frac{1}{2}} \tag{5.7}$$

As introduced in Chapter 3, the CR bound for such an overall performance measure is given by

$$\overline{rmse}(\mathbf{f}) \quad \geq \quad T(\mathbf{f}) \tag{5.8}$$

where $\overline{rmse}$ was defined by equation (3.7). This performance measure offers a bound in units of gray levels.

Similarly, an overall measure of registration performance for the set of unknown translations is given by the average MSE in estimating the entire set of unknown motion vectors $\vec{v}$. We denote the bound on such a performance measure by

$$T(\vec{v}) \quad = \quad \left(\frac{tr(\mathcal{S}_{\vec{v}}^{-1})}{2K}\right)^{\frac{1}{2}} \tag{5.9}$$

which gives the root average MSE error in estimating translation units of pixels over the given set of K unknown translations. The 2 comes from the 2 components of $\mathbf{v}_k$. The corresponding

CR inequality is given by

$$\overline{rmse}(\vec{\mathrm{v}}) \quad \geq \quad T(\vec{\mathrm{v}}) \tag{5.10}$$

It is the structure and behavior of these performance measures which we analyze in the following section.

5.3 Analysis of the CR Bound

In this section we explore the various aspects of the joint registration and restoration problem as it relates to the CR bound. Specifically, we study the complex relationship between image content, noise power, and motion vectors. We break down the analysis into the two subproblems of sub-Nyquist image registration and image reconstruction. For each subproblem, we first study the scenario where there is no available prior information about the unknown parameters, or $\mathbf{J}_p = 0$. Later, we study the effect additional prior information has on the estimation performance bounds for each subproblem. To simplify the presentation and convey the maximum intuition, we study the 1-D version of the problem. Where applicable, we denote the unknown motion for the 1-D case by the scalar translation parameter υ_k. We show in the appendix later that the extension to 2-D is straightforward.

Before we begin our analysis, we note that much of the analysis is simplified by examining the problem in the Fourier domain. Furthermore, many of the relevant matrices are diagonalized by the Fourier Transform allowing very efficient numerical implementations. To differentiate between the Fourier domain and spatial domains, we use a tilde as in$\widetilde{\mathbf{f}}$ to indicate vectors and matrices in the Fourier domain. Furthermore, we note that because the Discrete Fourier Transform (DFT) operation ($\widetilde{\mathbf{f}} = \Phi_{DFT}\mathbf{f}$) is a unitary transformation, the global bounds on image reconstruction (5.8) remains unchanged. In other words,

$$tr(\mathcal{S}_{\mathbf{f}}^{-1}) = tr(\Phi_{DFT}\mathcal{S}_{\mathbf{f}}^{-1}\Phi_{DFT}^{H}). \tag{5.11}$$

because Φ_{DFT} is a unitary operator [67]. This is basically Parseval's relation.

For the duration of this chapter, we assume that the unknown image $\mathbf{f}$ and the measured images $\mathbf{z}_k$ are both real-valued signals. Such a constraint induces symmetries in the frequency domain signal. Thus, while there are $2N_H$ coefficients (N_H real and N_H imaginary), we only need to estimate N_H of these components because of the symmetry. For instance, we define the problem as that of estimating the spectral coefficients in the positive frequency half-plane. Furthermore, to avoid the use of complex notation, we separate the real and imaginary components of the Fourier domain signal and stack them as in

$$\widetilde{\mathbf{f}} = \begin{pmatrix} Re\{\mathcal{F}(\theta_n)\} \\ Im\{\mathcal{F}(\theta_n)\} \end{pmatrix}, \ \theta_n = \frac{n2\pi}{N_H}, \ i = 0, \ldots, \frac{N_H}{2} \tag{5.12}$$

where $\mathcal{F}(\theta)$ is the DFT of the signal $f_0, f_1, \ldots$ (the components of $\mathbf{f}$). Here, the θ_n terms indicates the spatial frequencies comprising the signal $\mathbf{f}$. We note that the dimensions of the image vectors $\widetilde{\mathbf{f}}$ and $\widetilde{\mathbf{z}}$ are equal to those of their spatial counterparts $\mathbf{f}$ and $\mathbf{z}$.

The convolution operator $\mathbf{U}$ is block-diagonalized by the DFT. As such, the shift matrix $\widetilde{\mathbf{U}}_k$ is given by

$$\widetilde{\mathbf{U}}_k = \begin{pmatrix} diag(\cos(v_k\theta_n)) & -diag(\sin(v_k\theta_n)) \\ diag(\sin(v_k\theta_n)) & diag(\cos(v_k\theta_n)) \end{pmatrix} \tag{5.13}$$

Finally, the downsampling matrix $\mathbf{D}$ has the following structure

$$\widetilde{\mathbf{D}} = \frac{1}{M} \begin{pmatrix} \widetilde{\mathbf{D}}_R & 0 \\ 0 & \widetilde{\mathbf{D}}_I \end{pmatrix} \tag{5.14}$$

where

$$\{\widetilde{\mathbf{D}}_R\}_{i,j} = \begin{cases} 1, & i = j - 2aN_L, \ a = 0, 1, \ldots \\ 1, & i = 2aN_L - j, \ a = 1, 2, \ldots \\ 0, & else \end{cases} \tag{5.15}$$

$$\{\widetilde{\mathbf{D}}_I\}_{i,j} = \begin{cases} 1, & i = j - 2aN_L, \ a = 0, 1, \ldots \\ -1, & i = 2aN_L - j, \ a = 1, 2, \ldots \\ 0, & else \end{cases} \tag{5.16}$$

This structure reflects the spectral 'folding' or aliasing due to the downsampling operation. Figure 5.1 shows an example image of the matrix $\widetilde{\mathbf{D}}$ for $M = 3$. We note that in the Fourier domain, the upsampling operation ($\mathbf{D}^T$ in the time domain) is noted $\widetilde{\mathbf{D}}^{\intercal}$ where

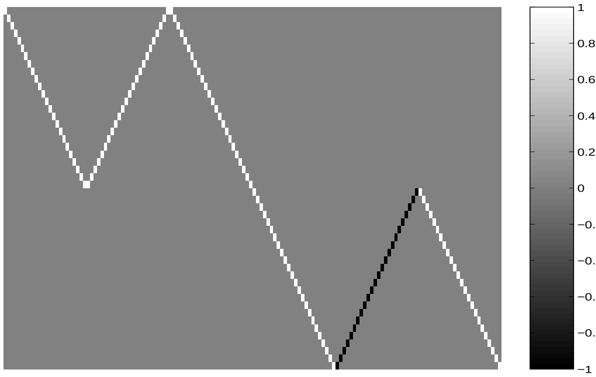

Figure 5.1: Image of the matrix $\widetilde{\mathbf{D}}$ for $M = 3$.

$$\widetilde{\mathbf{D}}^{\intercal} = \begin{pmatrix} \widetilde{\mathbf{D}}_R & 0 \\ 0 & \widetilde{\mathbf{D}}_I \end{pmatrix}^T \tag{5.17}$$

differing from $\widetilde{\mathbf{D}}$ only in that the weight $\frac{1}{M}$ is not used.

Under these assumptions, we can show that the terms of the FIM $\widetilde{\mathbf{J}}_d$ are given by

$$\begin{aligned}
\widetilde{\mathbf{J}}_{\mathbf{ff}} &= \frac{1}{\sigma^2}\sum_{k=0}^{K}\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^{\intercal}\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k = \frac{1}{\sigma^2}\sum_{k=0}^{K}\widetilde{\mathbf{Q}}(v_k) && (5.18)\\
\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathbf{v}}} &= \frac{1}{\sigma^2}[\cdots\ \widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^{\intercal}\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\boldsymbol{\Theta}\widetilde{\mathbf{f}}\ \cdots] = \frac{1}{\sigma^2}[\cdots\ \widetilde{\mathbf{Q}}(v_k)\boldsymbol{\Theta}\widetilde{\mathbf{f}}\ \cdots] && (5.19)\\
\widetilde{\mathbf{J}}_{\vec{\mathbf{v}}\vec{\mathbf{v}}} &= \frac{1}{\sigma^2}diag[\widetilde{\mathbf{f}}^T\boldsymbol{\Theta}^T\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^{\intercal}\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\boldsymbol{\Theta}\widetilde{\mathbf{f}}] = \frac{1}{\sigma^2}diag[\widetilde{\mathbf{f}}^T\boldsymbol{\Theta}^T\widetilde{\mathbf{Q}}(v_k)\boldsymbol{\Theta}\widetilde{\mathbf{f}}] && (5.20)
\end{aligned}$$

where the matrix $\boldsymbol{\Theta}$ arises from

$$\frac{\partial}{\partial v_k}\widetilde{\mathbf{U}}(v_k) = \widetilde{\mathbf{U}}(v_k)\begin{pmatrix} 0 & diag\{\theta_j\} \\ diag\{\theta_j\} & 0 \end{pmatrix} = \widetilde{\mathbf{U}}(v_k)\boldsymbol{\Theta} \tag{5.21}$$

The matrix $\boldsymbol{\Theta}$ corresponds to a derivative operation in the spatial domain. The derivation of these terms for the 2-D case can be found in Appendix 5.A. To simplify the notation, we represent the derivative signal by $\widetilde{\mathbf{d}}$ as in

$$\widetilde{\mathbf{d}} = \boldsymbol{\Theta}\widetilde{\mathbf{f}}$$

The matrix $\widetilde{\mathbf{D}}^{\mathsf{T}}\widetilde{\mathbf{D}}$ can be interpreted as a projection operator which maps the high resolution (dimension) image onto a lower dimensional measurement space. Simple calculations will verify that $\widetilde{\mathbf{D}}^{\mathsf{T}}\widetilde{\mathbf{D}} = \widetilde{\mathbf{D}}^{\mathsf{T}}\widetilde{\mathbf{D}}\widetilde{\mathbf{D}}^{\mathsf{T}}\widetilde{\mathbf{D}}$. Furthermore, because the operator $\widetilde{\mathbf{U}}_k$ is a unitary, $\widetilde{\mathbf{Q}}(v_k) = \widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^{\mathsf{T}}\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k$ is also a projection operator [67]. Finally, we note that the matrix $\widetilde{\mathbf{Q}}_k$ can be expressed as a linear combination by

$$\widetilde{\mathbf{Q}}_k \quad = \quad \frac{1}{M}\left(\mathbf{I} + \sum_{m=1}^{M-1}\left[\boldsymbol{\Lambda}_m^c \cos(m\phi_k) + \boldsymbol{\Lambda}_m^s \sin(m\phi_k)\right]\right) \tag{5.22}$$

where $\phi_k = \frac{v_k 2\pi}{M}$. The term ϕ_k can be thought of as the sampling phase offset for the kth measured low resolution image. This expansion is shown in Appendix 5.C. The matrices $\boldsymbol{\Lambda}_m$ are all symmetric matrices with zeros along the diagonal. They represent the portions of the folded spectrum due to the downsampling. As we shall soon show, the information content present in the signal is dependent on the sampling phase offset ϕ. Or, the Fisher Information is a periodic function of the the motion in the range $v \in [0, M]$. In the following sub-sections we analyze the CR bound matrices associated with subproblems of image reconstruction, registration, and restoration. As we shall show, the CR performance bounds associated with the image reconstruction problem possess a certain symmetric interdependence with the bounds on sub-Nyquist image registration.

5.3.1 Bounds on Registration of Aliased Images

In this section, we analyze the performance bound for the problem of registering aliased images. For this problem we must study the matrices $\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}$ and $\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^T\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}$ of (5.5). To date, the problem of registering aliased images has not been studied in relation to that of image reconstruction. As we will show in this section, if the images to be registered are sampled above the Nyquist rate, then image registration can be performed in a pairwise fashion and the registration performance is independent of image reconstruction. When the images are sampled below the Nyquist rate (hence are aliased), however, image registration and reconstruction are tightly coupled problems and they must be solved jointly using the entire set of images. In this section, we study the overall registration performance bound $T(\vec{\mathrm{v}})$, as measured by (5.9), and its relationship to the image reconstruction performance arriving at a general CR bound for aliased image registration.

We can learn much from the performance bound for sub-Nyquist image registration by looking at the first term of $\widetilde{\mathcal{S}}_{\vec{\mathrm{v}}}$, which is $\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}$. In fact, this term is the FIM for image registration when the high resolution image $\widetilde{\mathbf{f}}$ is known prior to estimation. As such, $\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}^{-1}$ offers an optimistic lower bound on sub-Nyquist registration performance. Because$\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}$ is a diagonal matrix, we can infer that, were the high resolution image $\widetilde{\mathbf{f}}$ known prior to estimation, the process of registering the measured images to the known high resolution image could be performed in a frame-by-frame fashion. Such an observation is consistent with the intuition that, if given the high resolution image $\widetilde{\mathbf{f}}$, one need not look at other low-resolution frames to register a particular low resolution image $\widetilde{\mathbf{z}}_k$.

In looking at the diagonal terms of the FIM sub-matrix $\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}$, we can see that the information for a particular frame $\{\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}\}_{kk}$, depends directly on the unknown translation v_k according to the function

$$\begin{aligned}\mathcal{I}(\phi_k,\widetilde{\mathbf{d}}) &\equiv \widetilde{\mathbf{d}}^T\widetilde{\mathbf{Q}}_k\widetilde{\mathbf{d}}\\ &= \frac{1}{M}\left(\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}+\sum_{m=1}^{M-1}\left[(\widetilde{\mathbf{d}}^T\boldsymbol{\Lambda}_m^c\widetilde{\mathbf{d}})\cos(m\phi_k)+(\widetilde{\mathbf{d}}^T\boldsymbol{\Lambda}_m^s\widetilde{\mathbf{d}})\sin(m\phi_k)\right]\right)\end{aligned}\tag{5.23}$$

In other words, the information necessary for registration depends on the energy in the spatial derivatives (texture) of the unknown signal $\widetilde{\mathbf{d}}$ projected into the lower dimensional measurement sub-space via the operator $\widetilde{\mathbf{Q}}_k$ defined in (5.18). We can make several observations about the information function $\mathcal{I}$ as it relates to the signal $\widetilde{\mathbf{f}}$, motion vectors v_k, and the downsampling factor M. First, recalling that $\widetilde{\mathbf{d}} = \boldsymbol{\Theta}\widetilde{\mathbf{f}}$, we see that any low pass filtering due to the blurring of the imaging system reduces the ability to register the images by damping the energy in the higher spatial frequencies. For example, let us suppose that, prior to capturing the image, the image function were to be blurred by a low-pass filter denoted $\widetilde{\mathbf{H}}$. Then, $\mathcal{I}(\phi,\boldsymbol{\Theta}\widetilde{\mathbf{f}}) \geq \mathcal{I}(\phi,\boldsymbol{\Theta}\widetilde{\mathbf{H}}\widetilde{\mathbf{f}})$ for all sampling phase offsets ϕ. This generalizes the observation introduced earlier in Chapter 3 that higher frequency information or texture improves the ability to register images. Naturally, the amount of information lost due to the low pass filter $\widetilde{\mathbf{H}}$ depends on the spectral content of the image $\widetilde{\mathbf{f}}$. Second, we now see the periodicity of the Fisher Information $\mathcal{I}$ as a function of v with a period of M. For the super-Nyquist case studied in Chapter 3, the information matrix was shown to be independent of the unknown translation v. Because of the downsampling operation $\widetilde{\mathbf{D}}$, this observation no longer holds for sub-Nyquist registration. In general, the information lost due to the downsampling operation can be quite significant.

As a first approximation, the downsampling operation alone reduces the information on the order of $\frac{1}{M}$. For example, Figure 5.2 shows the value of $\mathcal{I}(\phi, \widetilde{\mathbf{d}})$ of (5.23) throughout the range of sample phase offsets ϕ using the signal $\widetilde{\mathbf{f}}$ shown in Figure 5.6. The function is shown in polar coordinates about the angle ϕ. Immediately, we see that the information can vary quite dramatically for different sampling offsets ϕ. Because the performance bound can vary so widely for different values of ϕ, it is important to explore the entire space of translations v when performing simulation-based experiments. We note that this has generally not been the practice in the past.

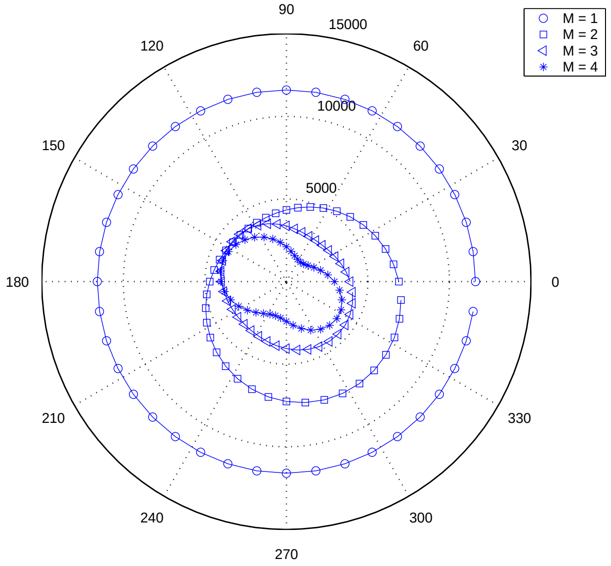

Figure 5.2: Polar plot of $\mathcal{I}(\phi, \widetilde{\mathbf{d}})$ (in units of $\frac{gray\ levels^2}{pixel^2}$) verses ϕ (in degrees) for different downsampling factors.

Perhaps more common is the scenario where the estimator has no knowledge of the high-resolution ”reference” image prior to registration of the low resolution images. In this situation, the information loss due to uncertainty about the high-resolution image is captured by the term $\widetilde{\mathbf{J}}^T_{\mathbf{f}\vec{v}}\widetilde{\mathbf{J}}^{-1}_{\mathbf{ff}}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}$. By way of the matrix inversion lemma [67], we see that the complete performance bound is given by

$$\widetilde{\mathcal{S}}_{\vec{v}}^{-1} = \widetilde{\mathbf{J}}^{-1}_{\vec{v}\vec{v}} + \widetilde{\mathbf{J}}^{-1}_{\vec{v}\vec{v}}\widetilde{\mathbf{J}}^T_{\mathbf{f}\vec{v}}\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}\widetilde{\mathbf{J}}^{-1}_{\vec{v}\vec{v}} \tag{5.24}$$

From this equation, we see that the performance bound for sub-Nyquist image registration is dependent on the image reconstruction performance bound $\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}$. Consequently, if the set of translations is such that the image reconstruction is ill-posed (or $\widetilde{\mathcal{S}}_{\mathbf{f}}$ is singular), then the corresponding problem of image registration is ill-posed as well. In other words, if the signal $\widetilde{\mathbf{f}}$ could not be reconstructed even if the sampling phase offsets were known perfectly (hence $\widetilde{\mathcal{S}}_{\mathbf{f}}$ is singular), then the problem of registering the aliased images is singular as well. When the Fisher Information matrix is singular, any unbiased estimator of the set of translations will necessarily have infinite variance [69]. This raises a very important point concerning the canonical experimentation scenario used in [43, 70]. For such experiments, an image is downsampled according to the forward model (5.2) and experiments which perform pair-wise image registration are presented. For 2-D images, a pair of images is always insufficient to reconstruct the underlying high-resolution image. As such, these experiments fail to acknowledge the implicit bias which necessarily accompanies such algorithms. Without prior information about the unknown image, unbiased pairwise registration of aliased images is impossible.

In fact, even when M measured images are available with the translations falling perfectly onto the high resolution sampling *grid*, the problem of multiframe image registration is still ill-posed. While such image measurements can be shown to be optimal from a signal reconstruction perspective (assuming the translations were known) [71], when the translations must also be estimated from the data, the Fisher Information is singular. This is proved in Appendix 5.B. Furthermore, it can be shown that the FIM is singular whenever the offset differences modulo M, $\{mod_M(v_k - v_j)|\forall i, j\}$ have less than $M + 1$ *unique* elements. Furthermore, the FIM becomes very ill-conditioned when the offset vectors are even ’near’ singular regions. For example, Figure 5.3 shows a surface plot of the CR bound $T(\vec{v})$ for a downsampling fact $M = 2$ when three measured frames are available ($K = 2$) for the signal in Figure 5.6. We see that near the boundaries $v_{1,2} = v_0$, and along the line $v_1 = v_2$, the CR bound goes to infinity (the values are cropped for display). The performance bound exhibits similar behavior when extended to higher dimensions. Here, we see that performance bound for equally spaced translations $v_1 = \frac{2}{3}, v_2 = \frac{4}{3}$ falls within the *well* of the performance bound plot. Thus, while equally spaced translations may not offer the best set of translations, it ensures that the performance bound does not exhibit the singular behavior. We note that such singular behavior is independent of the signal under observation. Furthermore, as we will show later in Section 5.4, such singular behavior can be mitigated with prior information. In this section, however, we assume that the additional information comes from an additional low-resolution measurement at a unique offset. This guarantees that the Fisher Information will not be singular.

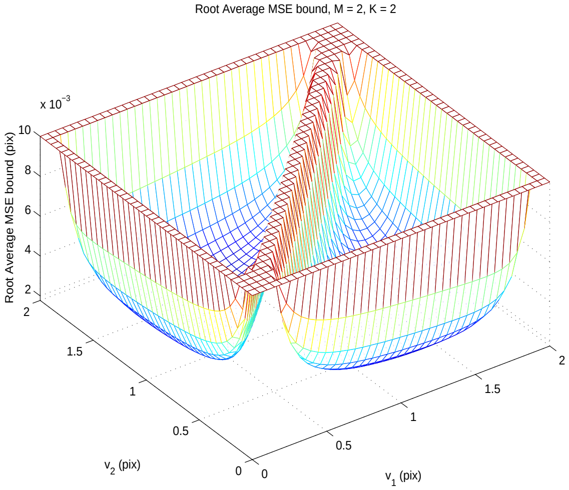

Figure 5.3: Plot of $T(\vec{v})$ for the signal of Figure 5.6 with $M = 2, K = 2$.

Because of the complicated structure of the CR bound, henceforth we compute the bound numerically for a given signal, translations, and noise power. For example, Figure 5.4 shows the overall performance bound $T(\vec{v})$, over the set of unknown motions for the signal shown in Figure 5.6. Each point in the plot indicates the performance bound for a set of $K + 1$

frames with equally spaced translations in the range $[0, M]$ assuming noise power $\sigma^2 = 1$. We note that increasing the number of frames does not affect the performance bound for the super-Nyquist scenario when $M = 1$. This suggests that an algorithm that performs pairwise registration could conceivably work as well as a more complicated algorithm which estimates a set of registration parameters using a set of low resolution images. This is not the case when the low resolution images contain aliasing. For downsampling factors greater than $M = 1$, we see that increasing the number of measured frames improves the overall performance bound. In some cases, the presence of additional frames cuts the overall performance bound in half. We can interpret this to mean that optimal sub-Nyquist image registration algorithms must estimate the set of translations $\{v_k\}$ from a set of low resolution measurements in a joint fashion. Estimating translations in using subsets of the collection of measured images $\{\tilde{\mathbf{z}}_k\}$ will necessarily result in a poorer performance bound. We shall see an example of such performance degradation in our experiments section.

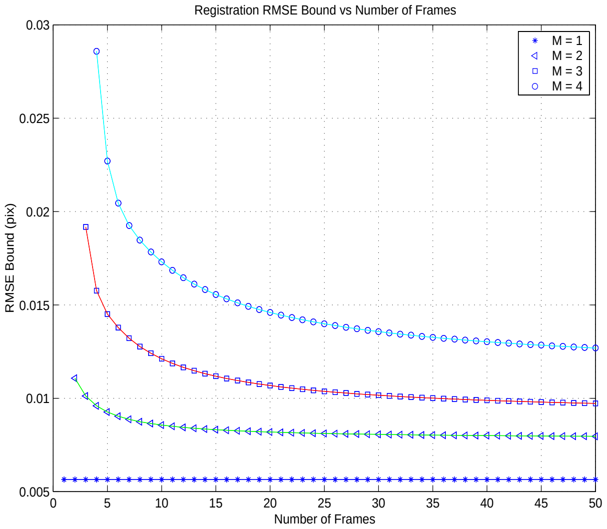

Figure 5.4: Overall Registration RMSE bound $T(\vec{\mathrm{v}})$ for equally spaced translations.

Ideally, we would like to study the performance bounds as a function of these offsets as they deviate from the equally-spaced scenario. Because of the difficulty in characterizing this explicitly, instead we study the average performance bounds when the offsets are drawn from a uniform distribution over the range $[0, M]$. In doing so, we can observe the qualitative behavior of the bounds as it relates to the set of translations $\{v_k\}$. For example, the Figure 5.5 shows $T(\vec{v})$ for 1000 sets of such random translations as dashed curves (shown in log scale) for the downsampling factors $M = 2, 3, 4$. For comparison, the dark-dashed lines represent the performance bounds for the equally spaced motions and the dark-solid lines represents the average of $T(\vec{v})$ over the 1000 random sets of translations. We can make several observations about the relationship between the performance bounds and the set of translations. First, we observe that the performance bound can be significantly worse for random offsets than for equally spaced offsets. While the random translations can sometimes offer slightly improved performance bounds, the equally spaced offsets provides a good approximation to the overall performance bound. Second, as the number of frames increases the average performance bound for the random offsets approaches the bound for the equally spaced offsets. This suggests that were the translations actually drawn from a random distribution, as the number of frames increases, we can reasonably expect the performance bound to be approximated by the equally spaced translations bound. Third, we see that the performance bound seems to flatten out suggesting that after a certain point, additional frames do not improve the performance bound for

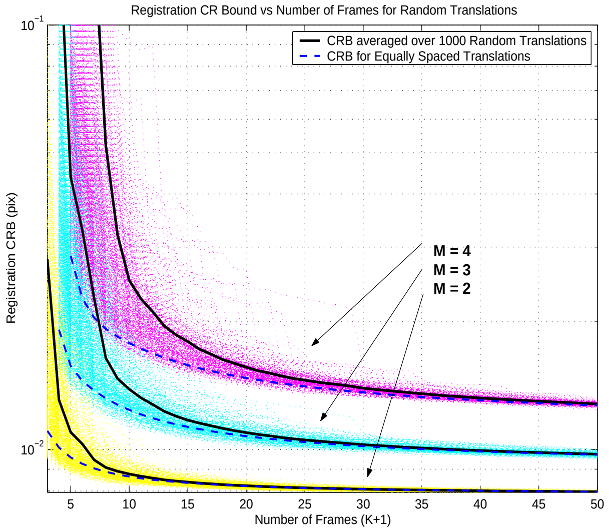

Figure 5.5: Registration CR bound $T(\vec{v})$ vs number of frames $K + 1$ for randomly selected translations.

sub-Nyquist registration.

5.3.2 Bounds on Image Reconstruction

As we saw earlier, the bounds on image registration depend, in part, on the performance bounds in reconstructing the image $\widetilde{\mathbf{f}}$. Therefore, we now study the performance bounds for the problem of multi-frame image reconstruction. To study the performance bounds associated with image reconstruction, we must analyze the matrix $\widetilde{\mathcal{S}}_{\mathbf{f}}$ of equation (5.6) in the context of image reconstruction.

As in the last section, the first term of $\widetilde{\mathcal{S}}_{\mathbf{f}}$, which is $\widetilde{\mathbf{J}}_{\mathbf{ff}}$, reflects the available information for image reconstruction when the estimator has full knowledge of the translation parameters prior to reconstruction. In fact, $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ is the FIM for such a scenario. Correspondingly, $\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}$ offers an overly optimistic bound on the more general problem of image reconstruction when the translation parameters must be estimated from the data. Another way to see this is by noting that

$$\widetilde{\mathbf{J}}_{\mathbf{ff}} \quad \geq \quad \widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^{T} \tag{5.25}$$

in the sense that the difference between these two terms is a positive semi-definite matrix [67]. From this, we see that $\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1} \leq \widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}$. Even as a weaker lower bound, much can be learned about the problem through analysis of $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ by itself.

When analyzing the problem in the presence of aliasing ($M > 1$), we can interpret the matrix $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ as a generalization of accumulating the *amount* of measurements for each high resolution pixel. We use the term *amount* rather than the number of measurements because when the sampling offset falls in between two *grid points* (i.e. not an integer), the pixel measurement is spread across the local pixels. By grid points, we refer to the common terminology used to describe the M, (or $M \times M$ for 2-D), sample locations corresponding to integer shifts of the high resolution image. This observation about $\mathbf{J}_{\mathbf{ff}}$ has been noted for the case of integer motion in [72]. Because of this, we expect the performance bound to vary spatially depending on the set of translations. In this section, we do not address the concern that$\widetilde{\mathbf{J}}_{\mathbf{ff}}$ might not be full rank. The condition number of $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ is related to the performance of signal restoration from interlaced sampling. This problem has been well studied in the signal processing community. For instance, [71] analyzes the stability of restoration for a given set of sampling offsets. It is interesting to note that the authors show that the ideal sampling offsets (assuming the offsets are known perfectly) corresponds to integer translations. We now show the more general property that $tr(\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1})$ is minimized by equally spaced motions $v \in [0, M]$ (not necessarily integer motion). To see this we use the matrix inequality

$$tr(\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}) \geq \sum_i \frac{1}{\{\widetilde{\mathbf{J}}_{\mathbf{ff}}\}_{ii}} \tag{5.26}$$

which applies for all matrices $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ which are symmetric [73]. Thus,

$$tr(\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}) \quad \geq \quad \sum_i \frac{1}{\{\widetilde{\mathbf{J}}_{\mathbf{ff}}\}_{ii}} = \frac{\sigma^2 N_H M}{K+1} \tag{5.27}$$

Next, we note that the matrix $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ for equally spaced motions is given by

$$\begin{aligned} \widetilde{\mathbf{J}}_{\mathbf{ff}} &= \frac{1}{M}\sum_{k=0}^{K}\left(\mathbf{I} + \sum_{m=1}^{M-1}\left[\mathbf{\Lambda}_m^c \cos(m\phi_k) + \mathbf{\Lambda}_m^s \sin(m\phi_k)\right]\right) \\ &= \frac{1}{M}\left((K+1)\mathbf{I} + \sum_{m=1}^{M-1}\sum_{k=0}^{K}\left[\mathbf{\Lambda}_m^c \cos(m\phi_k) + \mathbf{\Lambda}_m^s \sin(m\phi_k)\right]\right) \\ &= \frac{K+1}{\sigma^2 M}\mathbf{I}. \end{aligned} \tag{5.28}$$

where the trigonometric terms cancel since the motions are equally spaced. In other words, the trigonometric sums are of the form $\sum_{k=0}^{K} \cos(\frac{2\pi k}{K+1}) = \sum_{k=0}^{K} \sin(\frac{2\pi k}{K+1}) = 0$. As such, $tr(\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}) = \frac{\sigma^2 N_H M}{K+1}$ showing that equally spaced translations, assuming they known prior to image reconstruction, matches the weak lower bound (5.27). Thus, the set of all equally spaced motions achieves the bound on image reconstruction performance. Uniformly or equally spaced translations arise naturally if we assume that the image measurements are taken as samples in time of a scene with constant motion $v(t) = ct$ sampled uniformly at $t = kT$. Furthermore, it is not unreasonable to assume that if the images are samples from a scene whose motion is very slowly varying, for short periods of time, and high frame rates, the motion model is approximately constant.

To begin looking at the more general case, where the translations are not known a priori, we analyze the simple scenario where $M = 1$, or no downsampling (and hence no aliasing). The performance bound for this case characterizes the general behavior of the performance bound for $M > 1$. By way of the matrix-inversion lemma [67], we see that the general inverse $\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}$ can be written as

$$\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1} = \widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1} + \widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\tilde{\mathbf{v}}}\widetilde{\mathcal{S}}_{\tilde{\mathbf{v}}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\tilde{\mathbf{v}}}^{T}\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1} \tag{5.29}$$

(Here again, we see that $\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}$ is a weak lower bound on reconstruction performance.) When $M = 1$, we have

$$\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1} = \frac{1}{K+1}\mathbf{I} + \frac{K}{(K+1)}\frac{\widetilde{\mathbf{d}}\widetilde{\mathbf{d}}^{T}}{\widetilde{\mathbf{d}}^{T}\widetilde{\mathbf{d}}} \tag{5.30}$$

In this case, we see that for $M = 1$, the form for $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ is independent of the translations. The second term is a rank 1 matrix composed of outer product of the spatial derivative signal $\widetilde{\mathbf{d}}$. Such a term reflects the idea that image reconstruction (and later restoration) is more difficult in the textured regions. Essentially, this reflects the intuitive observation that errors in motion estimation will be most detrimental to image restoration in highly textured or high spatial frequency areas. For example, poor registration during multi-frame image reconstruction causes an edge-like feature to be distorted, creating sawtooth type artifacts [63]. This presents an interesting tradeoff in that the very image content which is easiest to register (highly textured) is also the content which is most prone to errors in the reconstruction. The full derivation of $\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}$ for the case $M = 1$ can be found in Appendix 5.D.

When $M > 1$, the second term $\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\tilde{\mathbf{v}}}\widetilde{\mathcal{S}}_{\tilde{\mathbf{v}}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\tilde{\mathbf{v}}}^{T}\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}$ adds uncertainty in the regions with large spatial derivatives. The graphs of Figure 5.7 shows the variance bound (diagonal of $\mathcal{S}_{\mathbf{f}}^{-1}$) for estimating the coefficient for each pixel/frequency for the signal shown in the graphs of Figure 5.6. The bound was calculated assuming four measured low resolution images with the translations $\{0.5,\ 1,\ 2\}$ and the reference frame, a downsampling factor of $M = 3$ and noise

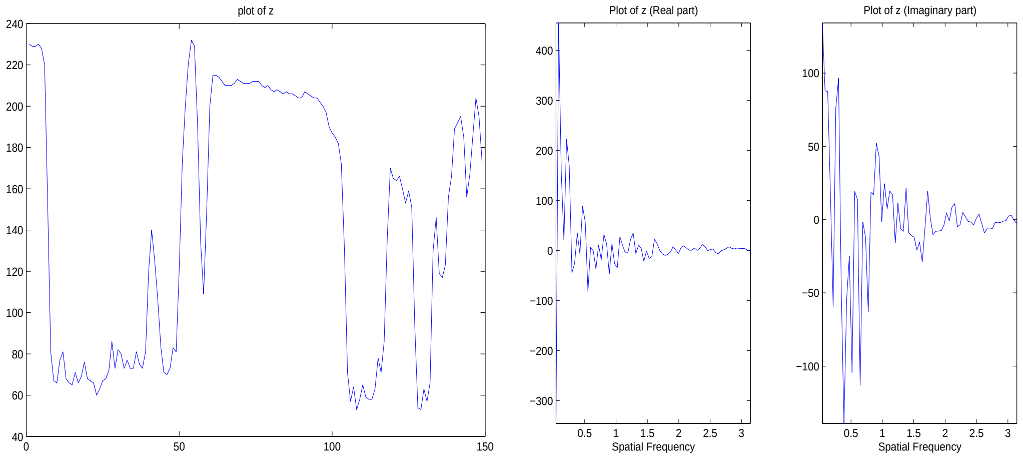

Figure 5.6: Plot of the signal $\mathbf{f}$ (left) and its separated Fourier Transform $\widetilde{\mathbf{f}}$ (right).

power $\sigma^2 = 1$. Here, we show the bound in the spatial domain to simplify its interpretation. The per-pixel variance bound has two distinct characteristics. First, the sawtooth-like periodic function comes from $\mathbf{J}_{\mathbf{ff}}^{-1}$ which reflects the amount of measured data associated with each pixel location in the high resolution image. This term is independent of the signal $\mathbf{f}$ and depends only on the number and the offsets of the low resolution images. Second, the spikes in performance bound arise from the $\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}\widetilde{\mathcal{S}}_{\vec{\mathrm{v}}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^{T}\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}$ term. Note that the spikes in the bound correspond to the locations of the 'edges' or high-frequency detail in the original spatial domain signal $\mathbf{f}$.

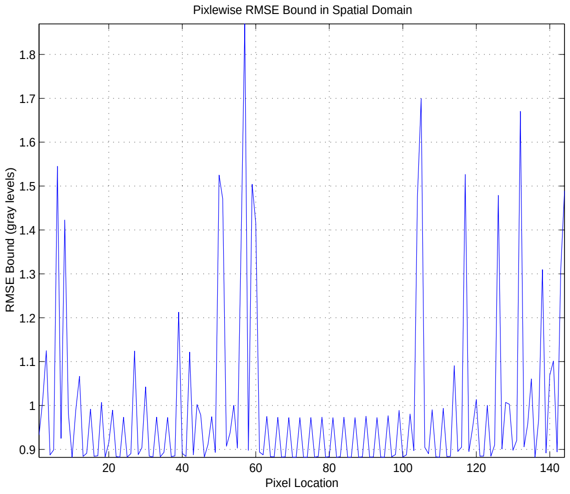

Figure 5.7: Variance bounds on image reconstruction shown for every pixel.

We can obtain a weak lower bound on the overall reconstruction performance using the inequality

$$Tr(\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}) \quad \geq \quad \frac{N_H^2}{Tr(\widetilde{\mathcal{S}}_{\mathbf{f}})}$$

from [68]. From this, we can bound on the RMSE bound $T(\widetilde{\mathbf{f}})$ by

$$\begin{aligned} T(\widetilde{\mathbf{f}}) \quad &\geq \quad \sigma\left(\frac{N_H}{Tr(\widetilde{\mathbf{J}}_{\mathbf{ff}}) - Tr(\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}\widetilde{\mathbf{J}}_{\vec{v}\vec{v}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}^{T})}\right)^{\frac{1}{2}} \\ &= \quad \sigma\left(\frac{N_H}{\frac{N_H(K+1)}{M} - K}\right)^{\frac{1}{2}} = \sigma\left(\frac{MN_H}{N_H(K+1) - KM}\right)^{\frac{1}{2}} \\ &= \quad \sigma\left(\frac{MN_H}{N_H(K+1) - KM}\right)^{\frac{1}{2}} \qquad (5.31) \end{aligned}$$

To compare this bound with the actual performance bound as the number of frames increases, we computed $T(\widetilde{\mathbf{f}})$ for the signal shown in Figure 5.6 assuming $\sigma^2 = 1$. In Figure 5.8, symbols show the value of $T(\widetilde{\mathbf{f}})$ for equally-spaced offsets. The solid lines indicate the weaker bound of (5.31). We note that the generic bound is fairly accurate, but seems to weaken as the downsampling factor M increases. This furthers the idea that equally spaced motions are nearly ideal for the problem of image reconstruction. We note that the weak bound suggests that the performance of image reconstruction depends primarily on the number of frames available.

Finally, to understand the sensitivity of the performance bound on the set of motion vectors $\vec{v}$ on the overall reconstruction performance bound, we compute the performance bound for randomly selected motions. In other words, we compute the value $T(\widetilde{\mathbf{f}})$ for the signal in Figure (5.6) for motion vectors drawn uniformly in the range $[0, M]$. Figure 5.9 shows the computed performance bound $T(\widetilde{\mathbf{f}})$ for these randomly drawn translations as the cloud of points. The solid line indicates the average of $T(\widetilde{\mathbf{f}})$ over the random set for each value of $K + 1$. As a point of reference, the dashed lines indicate the bound $T(\widetilde{\mathbf{f}})$ for the equally spaced translations. While Figure 5.9 does not offer insight into the functional relationship between $T(\widetilde{\mathbf{f}})$ and $\vec{v}$, it does show that as the number of frames increases, the variability of $T(\widetilde{\mathbf{f}})$ diminishes quite substantially. To summarize, if given a large enough collection of images

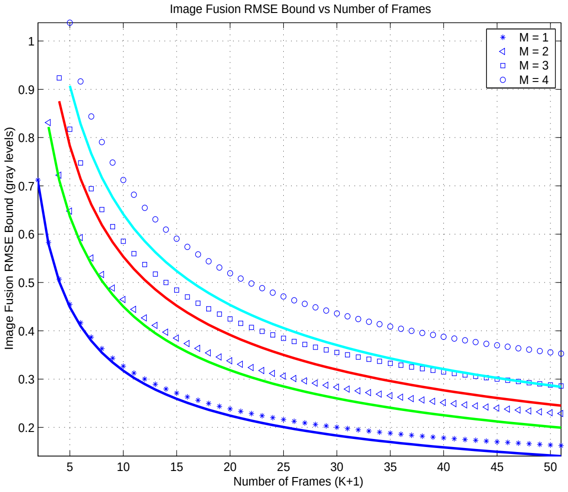

Figure 5.8: Plot of $T(\widetilde{\mathbf{f}})$ (symbols) and the weak bound approximation (5.31) (solid lines) vs $K+1$ for equally spaced translations.

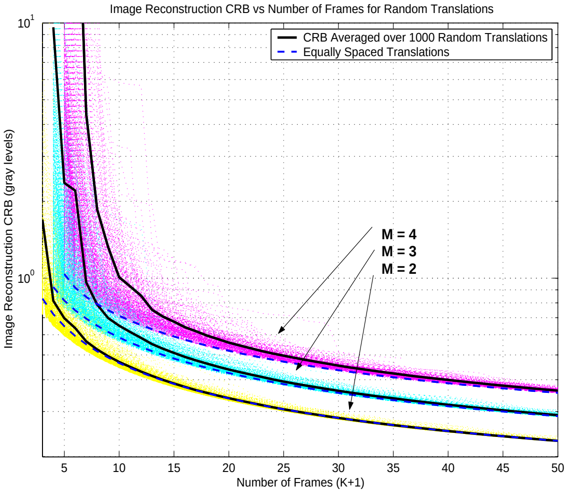

Figure 5.9: Scatter plot of $T(\mathbf{f})$ for random translations vs number of frames $K+1$.

with reasonably random offsets, the performance bound can be expected to be very close to the bound for equally spaced translations.

5.4 CR Bound with Prior Information

In practice, it is not uncommon to have some information about the unknown image prior to estimation. Such information is captured in the CR bound by the information term $\mathbf{J}_p$ in (3.5). Perhaps the most common form of prior information comes in the form of a Gaussian prior distribution over the space of unknown images $\widetilde{\mathbf{f}}$ [64, 72]. It suggests that the unknown image comes from the distribution $\widetilde{\mathbf{f}} \sim \mathcal{N}(\mu_{\widetilde{\mathbf{f}}}, \frac{1}{\lambda}\mathbf{C}_{\widetilde{\mathbf{f}}})$, where $\mu_{\widetilde{\mathbf{f}}}$ is the mean image with a covariance $\mathbf{C}_{\widetilde{\mathbf{f}}}$ and λ is a parameter capturing the overall confidence in the prior knowledge. In an algorithmic setting, often the λ term is used as a tuning parameter to control the strength of the prior information and hence its effect on the final estimate. From a statistical perspective, however, this term should reflect the *true* prior distribution. The Gaussian prior distribution has the probability density function

$$P(\widetilde{\mathbf{f}}) = \frac{1}{(\frac{2\pi}{\lambda})^{\frac{N}{2}}|\mathbf{C}_{\widetilde{\mathbf{f}}}|^{\frac{1}{2}}} \exp\left\{-\frac{\lambda}{2}(\widetilde{\mathbf{f}} - \mu_{\widetilde{\mathbf{f}}})^T \mathbf{C}_{\widetilde{\mathbf{f}}}^{-1}(\widetilde{\mathbf{f}} - \mu_{\widetilde{\mathbf{f}}})\right\} \tag{5.32}$$

Typically, the image distribution is assumed to have a diagonal covariance matrix $\mathbf{C}_{\widetilde{\mathbf{f}}}$ of the form

$$\mathbf{C}_{\widetilde{\mathbf{f}}} = \begin{pmatrix} diag(Re[\mathcal{X}(\theta_i)]) & 0 \\ 0 & diag(Im[\mathcal{X}(\theta_i)]) \end{pmatrix} \tag{5.33}$$

where $\mathcal{X}(\theta)$ is the the power spectral density for the image signal which is the Fourier transform of the image autocorrelation function. Prior information of this sort stems from some physical property relating to the functional smoothness of the image signal. In the Fourier domain, a natural measure of functional smoothness is given by

$$|\mathcal{X}(\theta)| \approx \frac{1}{|\theta|^\eta} \tag{5.34}$$

where η defines the global smoothness of the signal function [74]. Typically, the smoothness is chosen such that $\eta = 2$. The foundations of this prior information can be traced to physical properties inherent to natural scenes [52]. Such prior information can be interpreted to mean that the variability of the image signal diminishes for higher spatial frequencies.

Assuming that no prior information is available about the unknown translation parameters, the additional information provided by the prior on$\widetilde{\mathbf{f}}$ is given by

$$\mathbf{J}_p \;=\; \lambda \begin{pmatrix} \mathbf{C}_{\widetilde{\mathbf{f}}}^{-1} & 0 \\ 0 & 0 \end{pmatrix}^T \tag{5.35}$$

Interestingly, when $\eta = 2$, we see that $\mathbf{C}_{\widetilde{\mathbf{f}}}^{-1} = \mathbf{\Theta}^T\mathbf{\Theta}$. In other words, the statistical prior offers information about total energy in the first derivative of the signal. This form of prior information is commonly utilized in the literature to motivate the regularization penalty term in a Maximum A-Posteriori (MAP) estimator.

Typically, iterative super-resolution algorithms operating in the spatial domain use a finite impulse response (FIR) filter to approximate $\mathbf{C}_{\widetilde{\mathbf{f}}}^{-1}$, which for the case $\eta = 2$ turns out to be an FIR derivative filter. For example, perhaps the most common filter used to regularize the image estimates is the Laplacian approximation filter whose 1-D analogous filter is given by $[-1,\, 2,\, -1]$. In practice, higher order filter approximations can, but are rarely, used to more effectively incorporate prior information. Throughout the simulations which follow, we assume that the 1st order Laplacian filter approximation is used.

The prior information examined thus far is generic in the sense that it can be applied to a large class of images. Unfortunately, the generality of such prior information ultimately reduces its effectiveness in improving performance. Ideally, the practitioner of multi-frame image reconstruction and super-resolution may be able to ascertain more precise information when focusing on a particular application. In some situations, statistical properties about a certain class of images can be *learned* from large data sets providing very useful information. For instance, the authors of [75] show examples of incorporating learning-based priors into super-resolution for the particular restoration of facial and text images. It must be noted, however, that much care must be taken to ensure that training data sets are truly representative of the class of images for a particular application. Otherwise, the practitioner runs the very real risk of producing estimates heavily biased towards the training set. In many settings, a non-informative prior (suggesting higher variance) is safer than producing a biased estimate.

We now show the effect of such prior information on the respective bounds of image reconstruction and aliased image registration.

5.4.1 Prior Information and Image Registration Performance

In this section, we address the performance bound for image registration in the presence of aliasing under the assumption that prior information is available. Previous algorithms for sub-Nyquist registration implicitly incorporate prior information about the unknown signal. For instance, in [43], the authors make the observation that the effects of aliasing on measured image spectra is most prominent at high frequencies. As such, a generic algorithm for sub-Nyquist registration when $M = 2$ (or $M = 4$ for the 2-D scenario) is proposed which applies a nonlinear mask to the measured data prior to estimation to account for such aliasing effects. While such an algorithmic approach may indeed offer improved performance, the characterization of the prior information is ad hoc and needs to be quantified not only to understand general estimator performance, but also to derive efficient unbiased estimators.

Prior information about the unknown image can improve the performance bound for image registration, even in the event that no direct prior information is available about the unknown translations. Here, we focus on the performance bounds for the singular cases introduced in Section 5.3.1 and show how a prior on the image $\widetilde{\mathbf{f}}$, such as the Gaussian prior of (5.35), can significantly mitigate the singular behavior of the performance bound on motion estimation. For instance, in Figure 5.3 we showed the singular behavior of the CR bound for $M = 2$, $K = 2$ when the translations were *near* the singular set of translations. Correspondingly, the two graphs of Figure 5.10 show the same performance bound surface as Figure 5.3 with the addition of a Gaussian prior on the unknown image of the form (5.35) with $\lambda = .001$, and $\lambda = .01$. We see that the singular behavior near the integer motion shown in Figure 5.3 has been substantially diminished by the addition of prior information about $\widetilde{\mathbf{f}}$. Furthermore, we observe that the prior information does not affect the performance bounds away from the singularities. This suggests that if the motions were approximately equally spaced, little to no prior estimation is necessary

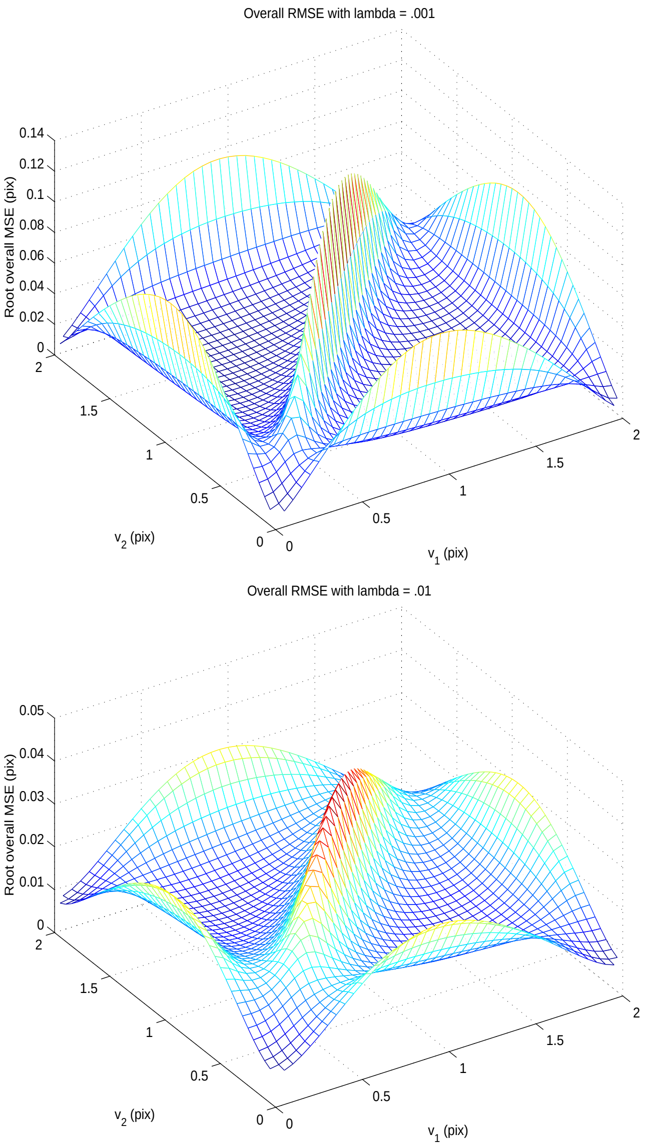

Figure 5.10: Surface plot of $T(\vec{v})$ with prior information for $M = 2, K = 2$.

to accurately register the images.

Next, we look at the effect of adding prior information while increasing the number of frames. Figure 5.11 shows the performance bounds averaged over the same 1000 random offsets from Figure 5.5, this time assuming differing amounts of prior information as parameterized by λ. Rather than show the point clouds of Figure 5.5, only the value of $T(\vec{v})$ averaged over the

1000 random translations for $M = 4$ is shown. As evidenced by Figure 5.5, when the translations are random, the performance bound tends to fluctuate wildly when only a few frames are available. For comparison, the faint lines show the performance bounds for the equally spaced translations for the same values of λ. When the translations are random, even small amounts of prior information substantially improves the stability of the performance bound when only a few frames are available. By stability, we refer to the fact that the average performance bound $T(\vec{v})$ over the set of 1000 random translations is much lower. Of course, when the translations are equally spaced, however, the problem is well conditioned and such small amounts of prior information does little to improve the performance bound.

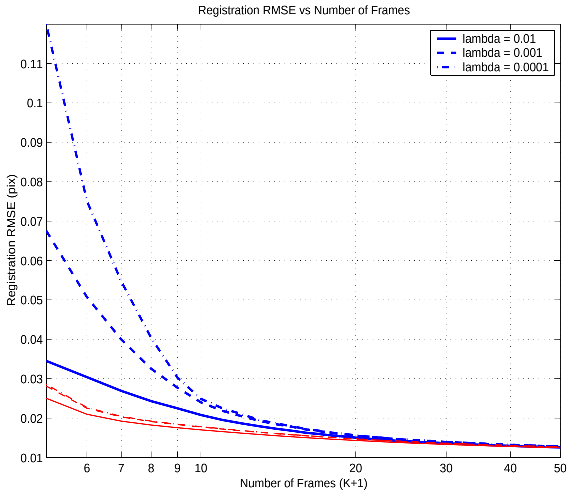

Figure 5.11: Registration CR bound for $M = 4$ with prior information vs number of frame $K + 1$ averaged over the set of 1000 random translations.

In the last section, we studied the optimistic performance bound $\widetilde{\mathbf{J}}^{-1}_{\vec{v}\vec{v}}$ which bounds performance where the image $\widetilde{\mathbf{f}}$ is known prior to estimation. When we have a prior on $\widetilde{\mathbf{f}}$, as the *strength* of prior information (in our case parameterized by λ) increases, the bound will approach this optimistic bound. For example, Figure 5.12 shows the performance bound for equally spaced translations versus the number of frames $K + 1$ as the λ goes from 0 (no prior information) to ∞ (perfect knowledge of the image $\widetilde{\mathbf{f}}$). The image function used in this experiments is that of Figure 5.6. Here, we see that as λ increases, the bound approaches that of $\widetilde{\mathbf{J}}^{-1}_{\vec{v}\vec{v}}$

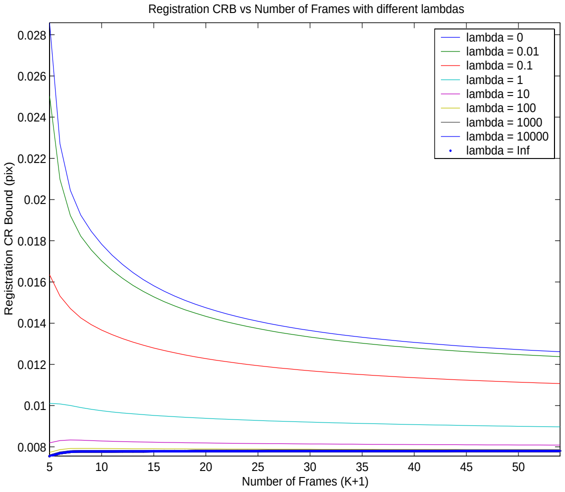

Figure 5.12: Registration CR bound for $M = 4$ with prior information vs number of frame $K + 1$ as λ goes from 0 to ∞.

(thick dotted line). When the prior information reaches this point, multiframe registration offers no improvement over pairwise estimation as evidenced by the flattened performance curve.

5.4.2 Prior Information and Image Reconstruction

As expected, prior information about the unknown image $\widetilde{\mathbf{f}}$ naturally offers information about $\widetilde{\mathbf{f}}$. Specifically, we see that a Gaussian prior on $\widetilde{\mathbf{f}}$ corresponds to a Gaussian prior information on $\widetilde{\mathbf{f}}$ according to

$$\widetilde{\mathbf{f}} \sim \mathcal{N}(\mu_{\widetilde{\mathbf{f}}}, \lambda \mathbf{C}_{\widetilde{\mathbf{f}}}). \tag{5.36}$$

Recalling that $\mathbf{J} = \mathbf{J}_d + \mathbf{J}_p$ we see that the only term which changes with the addition of prior information is $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ which is now given by

$$\widetilde{\mathbf{J}}_{\mathbf{ff}} = \frac{1}{\sigma^2}\sum_{k=0}^{K}\widetilde{\mathbf{Q}}(v_k) + \lambda \mathbf{C}_{\widetilde{\mathbf{f}}}^{-1} \tag{5.37}$$

In practice, such prior information mostly helps improve performance in the high frequency regions. For example, Figure 5.13 shows the CR bound on the variance per pixel (in the spatial domain) for the signal of Figure 5.6 with different amounts of prior information captured by λ. The bound was computed for $M = 3$ and $K = 4$ equally spaced motions (hence the lack of sawtooth behavior). It is apparent that the addition of prior information improves the performance bound in the flat regions somewhat, but has a much more significant effect at the edge locations. This improvement can be explained by both the additional knowledge about the high frequency content as well as improved registration performance.

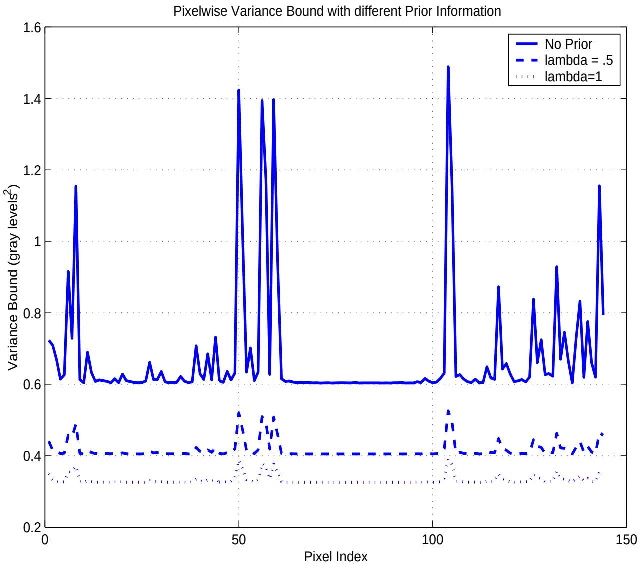

Figure 5.13: CR variance bound per pixel with different amounts of prior information.

5.5 Multiframe Image Registration Experiments

In this section, we compare the estimator performance of a standard multiscale gradient-based estimation algorithm as well as the aliased image registration algorithm [43] with the corresponding CR bounds on multi-frame aliased image registration. As we have shown previously, the standard gradient-based algorithm is designed to address non-aliased images. For our experiments, we used a 3 level multiscale algorithm with the Fleet gradient filter shown to offer reasonable performance in Chapter 4. Such an algorithm is expected to perform poorly in the presence of aliasing. The Stone et.al. algorithm [43], however, was specifically proposed to address the problem of registering a pair of aliased images. As we have shown, without prior information, such pairwise aliased image registration is ill-posed. In deriving the algorithm, the

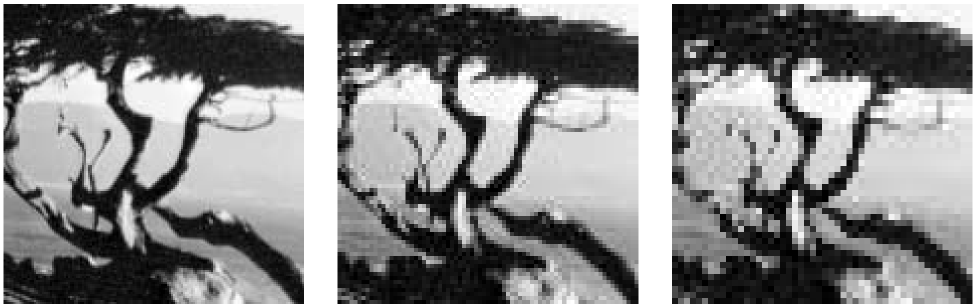

Figure 5.14: Tree image with no downsampling (left), $M = 2$ (middle) and $M = 3$ (right).

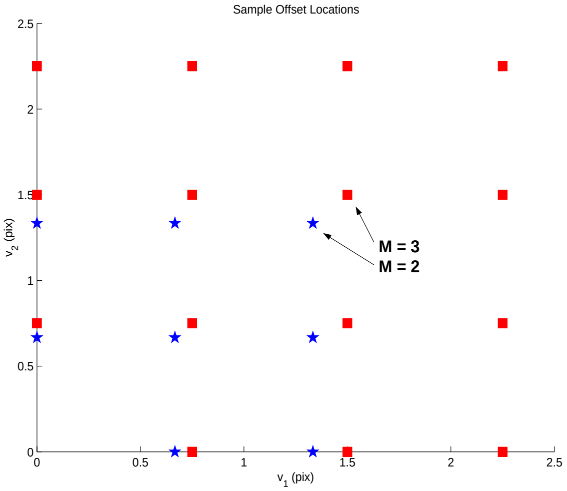

Figure 5.15: 2-D Equally-spaced translations for $M = 2$ and $M = 3$.

authors make several heuristic observations which they use to motivate the algorithm. In particular, the algorithm applies a nonlinear weighting of zeros and ones (a mask) to prune away portions of the image spectrum where the negative effects of aliasing are assumed to worsen estimation performance. For our experiments, we used parameter settings recommended in [43]. As we shall see, while such assumptions improve performance over the gradient-based algorithm, the algorithm's performance suggest significant room for improvement.

We perform our experiments using the Tree image shown in Figure 5.14. Figure 5.14 also shows an example of the Tree image downsampled by a factor of $M = 2$ and $M = 3$.

We conduct experiments using equally-spaced translation (in 2-D). In order that the estimation problem be well conditioned, we use $K + 1 = 8$ frames for $M = 2$ and $K + 1 = 16$

frames for $M = 3$. Figure 5.15 shows a scatterplot of the translation locations. Such offset locations guarantee that the FIM is well conditioned for both downsampling factors.

We evaluated the estimator performances for SNR values ranging from 20 to 60 db. Both registration algorithms were applied in a pair-wise fashion assuming the same reference frame. Figure 5.16 compares the performance of the two algorithms with the CR bound for the given set of images. Each point on the curve represents the value of $\overline{rmse}(\mathbf{v}_k)$ computed numerically for 500 MC runs. Here we see that the Stone algorithm outperforms the gradient-based algorithm at higher SNR's. We see that for SNR of 20 db, the gradient-based algorithm actually improves performance. This indicates that the statistical estimator bias balances out the deterministic bias associated with the gradient-based algorithm. Again, both algorithms show a flattening out of RMSE performance as SNR increases indicative of significant estimator bias. For a downsampling factor $M = 3$, the bias for both algorithms is greater than $\frac{1}{10}$ of a pixel. While such bias is highly dependent on the original image content, such estimator performance suggests that there is much work to be done in the area of aliased image registration. Overall, we conclude from these experiments that the current approaches to registering aliased images,

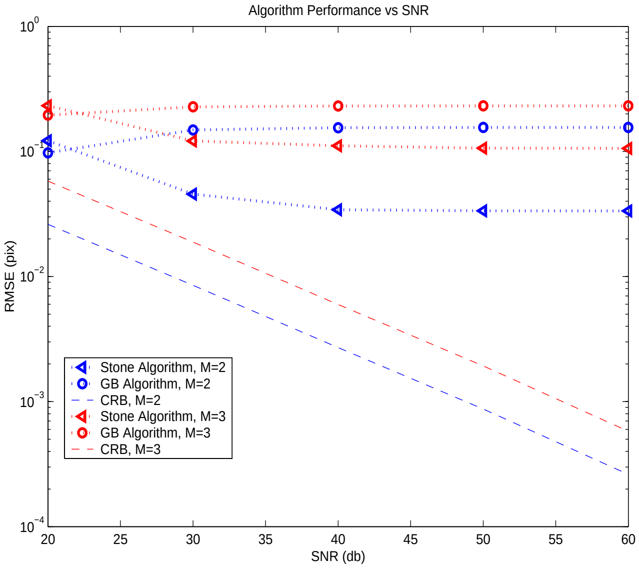

Figure 5.16: Experimental $\overline{rmse}(\vec{v})$ versus CR bound for $M = 2$ (blue) and $M = 3$ (red).

utilizing either a super-Nyquist algorithm or a heuristically designed sub-Nyquist algorithm, are inappropriate.

5.6 Conclusion

In this chapter, we have derived and explored the use of the Cramér-Rao inequality in bounding the performance for the joint problem of multiframe image reconstruction and aliased image registration. We have shown for the case of translational motion how the problem of registering aliased images naturally depends also on the subproblem of image reconstruction. We have analyzed the relationships between these two problems and characterized the performance limits of each. In addition, we outlined the importance of prior information in stabilizing the performance bound. Overall, the work has outlined several areas of research needing further attention. For instance, the problem of registering aliased images has been almost ignored as evidenced by the dearth of algorithms in the literature. Those that have looked at the problem, have approached the problem in a very ad-hoc fashion ignoring the fundamental relationship between image reconstruction and registration. Furthermore, to date the few algorithms addressing the problem of joint image registration and reconstruction have not addressed the problem from a proper estimation theoretic perspective. In our experimental section, the performance gap between the CR bound and the popular sub-Nyquist registration algorithm [43] revealing the need for further algorithmic development in the area of sub-Nyquist image registration. We will discuss this more in our final chapter. Finally, we note that much of the analysis of this chapter may be cross-applied to the problem of super-resolution.

5.A Fisher Information Matrix for the 2-D Scenario

In this appendix we show the necessary derivations for the 2-D version of the CR bounds for multi-frame image reconstruction and motion estimation. Recall that the modified forward model in the Fourier domain is

$$\widetilde{\mathbf{z}}_k = \widetilde{\mathbf{D}}\widetilde{\mathbf{U}}(\mathbf{v}_k)\widetilde{\mathbf{f}} + \widetilde{\mathbf{e}}_k.$$

The vector $\widetilde{\mathbf{f}}$ is a N_H dimensional vector with the first $\frac{N_H}{2}$ dimensions representing the real components and the the second $\frac{N_H}{2}$ dimensions representing the imaginary components. For the 1-D scenario, we used θ_i to identify the spatial frequency. For the 2-D scenario, we represent the spatial frequencies in the two dimensions as θ_1 and θ_2. For the 2-D scenario, all of the matrices have a similar structure as those of the 1-D scenario. Only the translation matrix $\widetilde{\mathbf{U}}(\mathbf{v}_k)$ is different in that the trigonometric terms are now are a function of the translation vector as $\cos(v_1\theta_{1_i} + v_2\theta_{2_j})$ and $\sin(v_1\theta_{1_i} + v_2\theta_{2_j})$.

The log-likelihood function for the observed data is given by

$$l(\{\widetilde{\mathbf{z}}_k\}|\widetilde{\mathbf{f}}, \vec{\mathrm{v}}) = \frac{-1}{2\sigma^2}\sum_{k=0}^{K}\left(\widetilde{\mathbf{z}}_k - \widetilde{\mathbf{D}}\widetilde{\mathbf{U}}(\mathbf{v}_k)\widetilde{\mathbf{f}}\right)^T\left(\widetilde{\mathbf{z}}_k - \widetilde{\mathbf{D}}\widetilde{\mathbf{U}}(\mathbf{v}_k)\widetilde{\mathbf{f}}\right)$$

Recal that the Fisher Information Matrix $\mathbf{J}$ is for such a problem is given by

$$J_{i,j} = -E\left[\frac{\partial^2 l(\{\widetilde{\mathbf{z}}_k\}|\widetilde{\mathbf{f}}, \vec{\mathrm{v}})}{\partial\psi_i\partial\psi_j}\right]$$

where ψ_i represents the particular parameter of interest. Computing these partial derivatives we see that

$$\begin{aligned}
-E\left[\frac{\partial^2 l(\{\widetilde{\mathbf{z}}_k\}|\widetilde{\mathbf{f}}, \vec{\mathrm{v}})}{\partial\widetilde{\mathbf{f}}^2}\right] &= \frac{1}{\sigma^2}\left[\sum_{k=0}^{K}\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^T\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\right] \\
&= \widetilde{\mathbf{J}}_{\mathbf{ff}}
\end{aligned}$$

and

$$\begin{aligned}
-E\left[\frac{\partial^2 l(\{\widetilde{\mathbf{z}}_k\}|\widetilde{\mathbf{f}}, \vec{\mathrm{v}})}{\partial\mathbf{v}_k^2}\right] &= \frac{1}{\sigma^2}\begin{bmatrix}\widetilde{\mathbf{f}}^T\mathbf{\Theta}_1^T\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^T\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\mathbf{\Theta}_1\widetilde{\mathbf{f}} & \widetilde{\mathbf{f}}^T\mathbf{\Theta}_1^T\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^T\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\mathbf{\Theta}_2\widetilde{\mathbf{f}} \\ \widetilde{\mathbf{f}}^T\mathbf{\Theta}_2^T\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^T\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\mathbf{\Theta}_1\widetilde{\mathbf{f}} & \widetilde{\mathbf{f}}^T\mathbf{\Theta}_2^T\widetilde{\mathbf{U}}_k^T\widetilde{\mathbf{D}}^T\widetilde{\mathbf{D}}\widetilde{\mathbf{U}}_k\mathbf{\Theta}_2\widetilde{\mathbf{f}}\end{bmatrix} \\
&= \left[\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}\right]_{kk}
\end{aligned}$$

where $\mathbf{\Theta}_{1,2}$ are the partial derivative operators in the Fourier domain which are block diagonal matrices of the form

$$\begin{aligned} \mathbf{\Theta}_1 &= \begin{pmatrix} 0 & -diag(\theta_1) \\ diag(\theta_1) & 0 \end{pmatrix} \\ \mathbf{\Theta}_2 &= \begin{pmatrix} 0 & -diag(\theta_2) \\ diag(\theta_2) & 0 \end{pmatrix} \end{aligned}$$

Finally, we see that,

$$\begin{aligned} -E\left[\frac{\partial^2 l(\{\widetilde{\mathbf{y}}_k\}|\widetilde{\mathbf{f}}, \vec{\mathrm{v}})}{\partial \mathbf{v}_k \partial \widetilde{\mathbf{f}}}\right] &= \frac{1}{\sigma^2}\left[\begin{array}{cc} \widetilde{\mathbf{U}}_k^T \widetilde{\mathbf{D}}^T \widetilde{\mathbf{D}} \widetilde{\mathbf{U}}_k \mathbf{\Theta}_1 \mathbf{z} & \widetilde{\mathbf{U}}_k^T \widetilde{\mathbf{D}}^T \widetilde{\mathbf{D}} \widetilde{\mathbf{U}}_k \mathbf{\Theta}_2 \widetilde{\mathbf{f}} \end{array}\right] \\ &= \widetilde{\mathbf{b}}_k \end{aligned}$$

So that our final FIM is given by

$$\mathbf{J}(\widetilde{\mathbf{f}}, \vec{\mathrm{v}}) = \begin{pmatrix} \widetilde{\mathbf{J}}_{\mathbf{ff}} & \widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}} \\ \widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^T & \widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}} \end{pmatrix}$$

where

$$\begin{aligned} \widetilde{\mathbf{J}}_{\mathbf{ff}} &= \sum_{k=0}^{K} \widetilde{\mathbf{U}}_k^T \widetilde{\mathbf{D}}^T \widetilde{\mathbf{D}} \widetilde{\mathbf{U}}_k \\ \widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}} &= [\cdots\ \widetilde{\mathbf{b}}_k\ \cdots] \\ \widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}} &= \begin{bmatrix} \left[\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}\right]_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \left[\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}}\right]_{KK} \end{bmatrix} \end{aligned}$$

5.B Singular FIM for Translations ”On the Grid”

In this appendix, we show that the Fisher Information matrix is necessarily singular when the set of translations $\{\mathbf{v}_k\}$ are all in units of whole pixels in the high resolution image. This corresponds to the canonical example in super-resolution experiments of having the low resolution frames falling perfectly on the ”grid” points. In this derivation, it is easier to conceptualize the proof in the spatial domain. Again, we show the proof for the 1-D case to simplify the presentation.

When the translations v_k are multiples of integer sample translations, the matrix $\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}$ is a diagonal matrix with the terms along the diagonal being $\frac{1}{K_i}$ where K_i represents the total number of low resolution frames with motions $v_k = v_i$ (corresponding to a particular grid location for the high-resolution image). This property has been noted in [72]. There are only M unique translations in the set of all translations and these translations are all integer offsets of the reference frame (in the high-resolution image coordinates). In other words, the motions are all on the super-resolution ”grid” points. We use A_i to denote the index set such that $v_k = v_i, \forall k \in A_i$. Without loss of generality, we assume that the unknown translations are ordered such that all $k \in A_i$ are contiguous. This ordering induces the structure on $\widetilde{\mathbf{J}}_{\vec{v}\vec{v}}$ such that

$$\widetilde{\mathbf{J}}_{\vec{v}\vec{v}} = \begin{pmatrix} c_0\mathbf{I}_{K_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & c_{M-1}\mathbf{I}_{K_{M-1}} \end{pmatrix} \tag{5.38}$$

where the subscript $\mathbf{I}_{K_i}$ indicates the dimension of the identity matrix. The coefficients are given by $c_i = \mathcal{I}(\phi_i, \mathbf{f})$. This ordering also induces structure on the matrix $\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}$ where the columns of $\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}$ which are associated with motions in the set A_i are all equal. Because of the structures of $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ and $\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}$, we see that $\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}^T\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}$ has a block diagonal form

$$\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}}^T\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{v}} = \begin{pmatrix} \mathbf{M}_0 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{M}_{M-1} \end{pmatrix} \tag{5.39}$$

where

$$\begin{aligned} \mathbf{M}_i &= \left(\frac{1}{K_i}\mathbf{f}^T\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{f}\right)\mathbf{1}\mathbf{1}^T \\ &= \left(\frac{c_i}{K_i}\right)\mathbf{1}\mathbf{1}^T \end{aligned} \tag{5.40}$$

where the last equality holds because $\mathbf{Q}$ is a projection operator and hence $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}$.

Thus, we see that the Schur complement Fisher Information is given by

$$\mathcal{S}_{\mathbf{f}} = \begin{pmatrix} \mathcal{S}_{\mathbf{f}_0} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathcal{S}_{\mathbf{f}_{M-1}} \end{pmatrix} \tag{5.41}$$

where

$$\mathcal{S}_{\mathbf{f}_i} = \begin{cases} c_i \left[\mathbf{I}_{K_i} - \frac{1}{K_i} \mathbf{1}\mathbf{1}^T \right], & i \neq 0 \\ c_i \left[\mathbf{I}_{K_i} - \frac{1}{K_i+1} \mathbf{1}\mathbf{1}^T \right], & else \end{cases} \tag{5.42}$$

which accounts for the fact that the first translation is assumed to be $v_0 = 0$ and is not an unknown. This shows that the for the very common scenario where the motions are in units of pixels, the information matrix is singular since $\mathcal{S}_{\mathbf{f}_i}$ is singular for $i \neq 0$. Each matrix $\mathcal{S}_{\mathbf{f}_i}$ is of rank $K_i - 1$ suggesting that the matrix $\mathcal{S}_{\mathbf{f}}$ is only rank deficient by $M - 1$.

5.C Decomposition of the Projection Operator $\widetilde{\mathbf{Q}}$

In this appendix, we study the projection operator $\widetilde{\mathbf{Q}}_k$. First, we note that

$$\widetilde{\mathbf{Q}}_k = \widetilde{\mathbf{U}}_k^T \widetilde{\mathbf{D}}^\dagger \widetilde{\mathbf{D}} \widetilde{\mathbf{U}}_k = \frac{1}{M} \begin{pmatrix} \widetilde{\mathbf{Q}}_{11}^k & \widetilde{\mathbf{Q}}_{12}^k \\ \widetilde{\mathbf{Q}}_{21}^k & \widetilde{\mathbf{Q}}_{22}^k \end{pmatrix} \tag{5.43}$$

Here, we show that there is a simplified $M \times M$ representation of the sub-matrices $\widetilde{\mathbf{Q}}_{ij}$. To see this, we note that

$$\widetilde{\mathbf{Q}}_{11}^k = \begin{pmatrix} \mathbf{I} & \cos(\phi_k)\mathbf{I}^F & \cos(\phi_k)\mathbf{I} & \cos(2\phi_k)\mathbf{I}^F & \dots \\ \cos(\phi_k)\mathbf{I}^F & \mathbf{I} & \cos(2\phi_k)\mathbf{I}^F & \cos(\phi_k)\mathbf{I} & \dots \\ \cos(\phi_k)\mathbf{I} & \cos(2\phi_k)\mathbf{I}^F & \mathbf{I} & \cos(3\phi_k)\mathbf{I}^F & \dots \\ \cos(2\phi_k)\mathbf{I}^F & \cos(\phi_k)\mathbf{I} & \cos(3\phi_k)\mathbf{I}^F & \mathbf{I} & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

where $\mathbf{I}^F$ represents the permutation matrix

$$\mathbf{I}^F = \begin{pmatrix} 0 & & & & 1 \\ & & & \cdot & \\ & & \cdot & & \\ & \cdot & & & \\ 1 & & & & 0 \end{pmatrix} \tag{5.44}$$

which when applied reverses the ordering of a vector. Thus, we can represent the matrices much more simply as

$$\mathbf{Q}^k_{11} = \begin{pmatrix} 1 & \cos(\phi_k) & \cos(\phi_k) & \cos(2\phi_k) & \dots \\ \cos(\phi_k) & 1 & \cos(2\phi_k) & \cos(\phi_k) & \dots \\ \cos(\phi_k) & \cos(2\phi_k) & 1 & \cos(3\phi_k) & \dots \\ \cos(2\phi_k) & \cos(\phi_k) & \cos(3\phi_k) & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The pattern for this is shown in Figure 5.17.

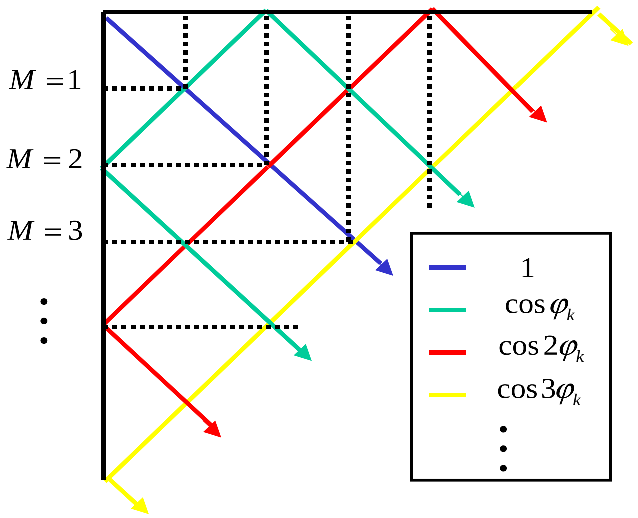

Figure 5.17: Pattern of the sub matrix $\widetilde{\mathbf{Q}}^k_{11}$.

Similarly, the other sub-matrices are given by

$$\tilde{\mathbf{Q}}_{22}^k = \begin{pmatrix} 1 & -\cos(\phi_k) & \cos(\phi_k) & -\cos(2\phi_k) & \dots \\ -\cos(\phi_k) & 1 & -\cos(2\phi_k) & \cos(\phi_k) & \dots \\ \cos(\phi_k) & -\cos(2\phi_k) & 1 & -\cos(3\phi_k) & \dots \\ -\cos(2\phi_k) & \cos(\phi_k) & -\cos(3\phi_k) & 1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\tilde{\mathbf{Q}}_{12}^k = \begin{pmatrix} 0 & -\sin(\phi_k) & -\sin(\phi_k) & -\sin(2\phi_k) & \dots \\ -\sin(\phi_k) & 0 & -\sin(2\phi_k) & -\sin(\phi_k) & \dots \\ \sin(\phi_k) & -\sin(2\phi_k) & 0 & -\sin(3\phi_k) & \dots \\ -\sin(2\phi_k) & \sin(\phi_k) & -\sin(3\phi_k) & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\tilde{\mathbf{Q}}_{21}^k = \begin{pmatrix} 0 & -\sin(\phi_k) & \sin(\phi_k) & -\sin(2\phi_k) & \dots \\ -\sin(\phi_k) & 0 & -\sin(2\phi_k) & \sin(\phi_k) & \dots \\ -\sin(\phi_k) & -\sin(2\phi_k) & 0 & -\sin(3\phi_k) & \dots \\ -\sin(2\phi_k) & -\sin(\phi_k) & -\sin(3\phi_k) & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

where $\phi_k = \frac{\pi v_k}{M}$

From this, we see that we can expand the matrix $\widetilde{\mathbf{Q}}_k$ as

$$\widetilde{\mathbf{Q}}_k = \frac{1}{M}\left(\mathbf{I} + \sum_{m=1}^{M-1} [\mathbf{\Lambda}_m^c \cos(m\phi_k) + \mathbf{\Lambda}_m^s \sin(m\phi_k)]\right) \tag{5.45}$$

where the terms $\mathbf{\Lambda}_m$ refer to the matrices of all ±1's denoting the locations of the trigonometric coefficients $\cos(m\phi_k)$ and $\sin(m\phi_k)$. Such an expansion will help us understand the behavior of the CR bounds.

5.D Derivation of the Schur Matrices for $M = 1$

Here we look at the case where there is no downsampling (just image fusion and registration). In this case we have that

$$\begin{aligned}
\widetilde{\mathbf{J}}_{\mathbf{ff}} &= (K+1)\mathbf{I} && (5.46)\\
\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}} &= [\cdots\ \boldsymbol{\Theta}\widetilde{\mathbf{f}}\ \cdots] && (5.47)\\
\widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}} &= \left(\widetilde{\mathbf{f}}^T\boldsymbol{\Theta}^T\boldsymbol{\Theta}\widetilde{\mathbf{f}}\right)\mathbf{I} = \left(\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}\right)\mathbf{I} && (5.48)
\end{aligned}$$

First, we note that the Schur complement of $\widetilde{\mathbf{J}}_{\mathbf{ff}}$ is given by

$$\begin{aligned}
\widetilde{\mathcal{S}}_{\vec{\mathrm{v}}} &= \widetilde{\mathbf{J}}_{\vec{\mathrm{v}}\vec{\mathrm{v}}} - \widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^T\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}\\
&= (\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}})\mathbf{I} - \frac{1}{K+1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^T\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}} && (5.49)\\
&= \widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}\left[\mathbf{I} - \frac{1}{K+1}\mathbf{1}\mathbf{1}^T\right] && (5.50)
\end{aligned}$$

Using the matrix inversion lemma [67], we see that

$$\begin{aligned}
\left[\mathbf{I} - \frac{1}{K+1}\mathbf{1}\mathbf{1}^T\right]^{-1} &= \mathbf{I} + \mathbf{1}\left(K+1-\mathbf{1}^T\mathbf{1}\right)^{-1}\mathbf{1}^T\\
&= \mathbf{I} + (K+1-K)^{-1}\,\mathbf{1}\mathbf{1}^T\\
&= \mathbf{I} + \mathbf{1}\mathbf{1}^T && (5.51)
\end{aligned}$$

where $\mathbf{1}$ represents a column vector of all ones of length K. So, the inverse of $\widetilde{\mathcal{S}}_{\vec{\mathrm{v}}}$ is given by (using the matrix inversion lemma)

$$\widetilde{\mathcal{S}}_{\vec{\mathrm{v}}}^{-1} = \frac{1}{\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}}(\mathbf{I} + \mathbf{1}\mathbf{1}^T) \qquad (5.52)$$

This has the same form as derived previously for looking only at the performance bounds for estimating translational motion [76]. Furthermore, it is interesting to note that for the case when no aliasing is present, adding additional frames to the problem does not influence the image registration problem. In other words, registration can be done in a pairwise fashion without any loss of information. This is not the case when aliasing occurs.

To capture the MSE performance in estimating the image terms $\widetilde{\mathbf{f}}$, we need only to look at the term

$$\begin{aligned}
\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1} &= \widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1} + \widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}\widetilde{\mathcal{S}}_{\mathbf{f}}^{-1}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^{T}\widetilde{\mathbf{J}}_{\mathbf{ff}}^{-1} \\
&= \frac{1}{K+1}\mathbf{I} + \frac{1}{(K+1)^2}\frac{1}{\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}}\left(\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^{T} + \widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}\mathbf{1}\mathbf{1}^T\widetilde{\mathbf{J}}_{\mathbf{f}\vec{\mathrm{v}}}^{T}\right) \\
&= \frac{1}{K+1}\mathbf{I} + \frac{K+K^2}{(K+1)^2}\frac{\widetilde{\mathbf{d}}\widetilde{\mathbf{d}}^T}{\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}} \\
&= \frac{1}{K+1}\mathbf{I} + \frac{K}{(K+1)}\frac{\widetilde{\mathbf{d}}\widetilde{\mathbf{d}}^T}{\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}}
\end{aligned}$$

Finally, we note that for this simple scenario the root average MSE bound as measured by (5.8) is given by

$$\begin{aligned}
T(\widetilde{\mathbf{f}}) &= \left(\frac{tr(\mathbf{I})}{N_H(K+1)} + \frac{K}{N_H(K+1)}\frac{\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}}{\widetilde{\mathbf{d}}^T\widetilde{\mathbf{d}}}\right)^{\frac{1}{2}} \\
&= \left(\frac{N_H+K}{N_H(K+1)}\right)^{\frac{1}{2}}
\end{aligned}$$

Chapter 6

Contributions and Future Work

This chapter summarizes the contributions made in the analysis of performance in motion estimation. We also detail several open questions related to this thesis as well as map out future research directions.

6.1 Contributions

In this thesis we studied general aspects of performance in estimating motion contained in image sequences. We constructed a well-defined description of the problem from an estimation theoretic point of view, allowing us to make foundational contributions to both the methodology and the science of motion estimation. We hope that our analytical framework will help guide and inform further advances in the wide array of fields that study and utilize motion estimation algorithms.

- In Chapter 2, we described a general theory regarding the use of tomographic projections to estimate motion. In particular, we presented the precise and approximate models of affine motion under tomographic projection. From this we showed a general scheme for estimating these affine motion parameters from a set of estimates of motion in the projected domain. Such concepts were presented in a general form so as to be agnostic regarding any particular algorithm for estimating the projected motion parameters. Specifically, we showed how to incorporate tomographic projections into a multiscale gradient-based algorithm for estimating affine motion. Such an algorithm was shown to achieve dramatic computational speedups while sacrificing little in the way of estimator accuracy for a wide range of operational scenarios.

- In Chapter 3, motivated by the interesting performance characteristics of the gradient-based algorithms, we posed the question of fundamental performance limits to motion estimation. Utilizing the Cramér-Rao bound, we explored these fundamental performance limits associated with translational motion estimation. We presented the experimental performance of several popular algorithms and compared their performance with the derived bound, showing the tendency for common algorithms to contain significant estimator bias.

- In Chapter 4, we focused on the class of gradient-based motion estimation algorithms. Motivated by the observations of estimator bias in Chapter 3, we derived a closed-form expression for the estimator bias for gradient-based algorithms. We verified that this bias expression indeed reflects estimator performance for high SNR scenarios and offered detailed analysis of the various components associated with this bias function. Using this bias formulation we constructed rule-of-thumb performance limits for the class of gradient-based algorithms. Also, from the bias formulation we proposed a novel method for improving algorithm performance for high SNR scenarios where the bias dominates performance.

- In Chapter 5, we extended our fundamental performance limits associated with image registration to the sub-Nyquist case showing the implicit relationship between sub-Nyquist registration and the problem of super-resolution. Our analysis offered new insight into the estimation theoretic challenges associated with the registration of aliased images, often revealing the implicit assumptions made by general practitioners. Finally, we proved the fundamental importance of prior information about the image function when attempting motion estimation in the sub-Nyquist scenario.

In closing, we note that much of the work presented in this thesis has resulted in several publications in peer reviewed journals and conferences [21, 34, 76, 77].

6.2 Future Work

In this section, we outline a few of the open questions related to the research presented in this thesis. In particular, we offer possible extensions to each of the chapters. Finally, we outline future areas of research which deserve attention.

6.2.1 Projection-Based Motion Estimation

The work presented in Chapter 2 explored a few of the many benefits of incorporating tomographic projections into motion estimation as an efficient mechanism for improving computational efficiency. In fact, we demonstrated that, in some instances, the projection-based estimation scheme offered improved performance. Here, we list several open questions and extensions to this work.

- We conjectured that the performance improvement/loss is highly dependent on the choice of projection angles. Further investigation into the choice of projection angles is warranted to maximize the possible performance for the projection-based estimators.

- In the field of gradient-based estimation, several robust estimators have been proposed over the years such as [38]. Such estimation techniques are much more computationally taxing than the traditional gradient-based algorithms but have been shown to offer improved performance under a wide variety of conditions. Because of their computational complexity, such algorithms would naturally benefit from the use of projections. It remains to be seen, however, if such robust projection-based estimators could achieve a similar improvement in performance while minimizing computational complexity.

- We believe that much of the analysis presented in this section will be useful to indirect imagine where only tomographic projections are measurable (magnetic resonance image, positron emission tomography etc). It would be interesting to explore the application of our projection-based estimators on such data.

6.2.2 Performance Analysis of Image Registration

There are several extensions to Chapter 3 that could prove extremely beneficial to the motion estimation community.

- The analysis presented in Chapter 3 focused on the simple case of translational estimation. One natural extension of this work is the examination of higher order motion models such the complete affine, bilinear, projective, etc. One would hope that detailed understanding of the performance bounds for such problems might illuminate the problem of model selection as it pertains to local estimation methods.

- Much of the performance analysis presented in Chapter 3 bears resemblance to the 1-D signal processing problem of delay estimation. It has been shown that for low SNR situations, the CR bounds begin to be overly optimistic for delay estimation. To address the performance bounds in these regions, other more sophisticated bounds such as the Ziv-Zakai bound [78], [79] and Barankin bounds [80, 81]. In certain applications where the SNR of the imaging system falls into this low SNR region, such bounds would be helpful in producing more realistic performance bounds.

- As we observed in Chapter 3, several translational estimation algorithms contain estimator bias. While the complete CR bound is capable of incorporating the bias term into the MSE bound, the bound becomes only applicable to the class of estimators with the same bias function. Recently, there have been several attempts to generalize the CR bounds for larger classes of biased algorithms whose bias gradients are constrained by some bound in what is called the Uniform CR bound [82]. Application of such bounds to the problem of image registration might offer insight into the fundamental bounds associated with general classes of biased estimators.

6.2.3 Gradient-Based Estimator Bias

In Chapter 4, we offered in-depth analysis of the bias structure associated with gradient-based translational estimation. Here, we offer a few general open questions related to this work.

- As before, one could imagine studying the bias properties of the gradient-based estimators for higher order parametric motion models. Finding such bias structures and employing a bias minimizing filter design approach could prove very useful for a large class of global image registration problems.

- The work on bias minimizing filters has several natural extension. One might derive the bias for lower SNR situations where the MSE is not dominated by bias. Thus, one could possibly design MSE optimal filters for gradient-based motion estimation. To do so, a functional characterization for the MSE at lower SNR must be developed. One simple approximation uses the CR bound itself as a cost function for optimizing the gradient filters.

- Much of the filter design process requires a reasonably accurate characterization of the image spectral content. Such characterization becomes difficult to obtain for local estimation with small windows. One possible research direction involves decomposing the gradient filter into a bank of filters each having well characterized bias structures. It might be possible to find an optimal locally adaptive linear combination of such filters which minimizes overall estimator bias.

6.2.4 Performance Analysis of Aliased Image Registration

Perhaps of all the chapters, Chapter 5 uncovers multiple areas for promising research.

General Open Research Questions

- Overall, the work in Chapter 5 uncovers the dearth of research into the area of motion estimation and image registration of aliased images. The application of superresolution requires such estimates making the analysis very relevant.

- The work presented in Chapter 5 may provide a foundation for systematic imaging system design where superresolution is known to be applied after capturing data. We imagine a scenario where engineering design decisions may be informed by the fundamental bound on image restoration.

- We note that a true Maximum Likelihood estimator for the joint problem of motion estimation and image restoration has not been addressed. Below, we detail future work related to such an estimator.

Maximum Likelihood Registration of Aliased Images

A natural question to ask when studying the CR bounds for a given estimation problem is wether an efficient estimator exists which can attain the given performance bounds. In general, this is an extremely difficult task, but it is well known that a Maximum Likelihood (ML) estimate is asymptotically efficient. In other words, as the number of measurements increases, the performance of the ML estimator approaches the CR bound. In this section, we show that finding the ML estimate for the joint image registration and reconstruction problem requires solving a nonlinear Least Squares (NLS) problem.

As noted in previous works such as [64, 72], the ML estimates for image reconstruction and registration minimize a cost function of the form

$$C_{ML}(\mathbf{f}, \vec{\mathrm{v}}) \quad = \quad \sum_k \|\mathbf{z}_k - \mathbf{D}\mathbf{U}_k\mathbf{f}\|_2^2 \tag{6.1}$$

Thus, we see that finding the ML solution requires minimizing a NLS cost function as both the set of motion vectors $\vec{\mathrm{v}}$ and the high resolution image $\mathbf{f}$ are unknown. Several approaches have been offered to minimizing such cost functions. Early work suggested that the estimation process could follow a two stage approach by first estimating the registration parameters between pairs of low resolution frames followed by minimization of a linear Least Squares (LLS) cost function to reconstruct the high resolution image. It was noted in [64], that such an algorithm often fails when the low resolution images contain significant aliasing artifacts due to sub-Nyquist sampling. In these situations, the image registration algorithms will almost assuredly provide biased estimates of the registration parameters. The authors in [64] correctly note that the proper approach must directly minimize the nonlinear cost function (6.1). They propose to do so using a cyclic coordinate descent algorithm where the algorithm cycles between the task of estimating the image **f** and the registration parameters $\{\mathbf{v}_k\}$, in each step assuming the other set is known. The authors also incorporate a prior on the unknown image **f** to improve the condition of the LLS problem. With such an algorithm, however, no assurance is given concerning the global convergence.

When examining the structure of (6.1), we see that the cost function is a special case of NLS where there exists a natural separability of the unknown parameters. In our case, we see that the data depend linearly on the unknown image **f** and nonlinearly on the registration parameters $\vec{\mathrm{v}}$. If we knew the registration parameters prior to image reconstruction problem, we see that the ML estimate of the image **f** is given by

$$\hat{\mathbf{f}} = \left[\sum_k \mathbf{U}_k^T \mathbf{D}^T \mathbf{D} \mathbf{U}_k\right]^{-1} \left[\sum_k \mathbf{U}_k^T \mathbf{D}^T \mathbf{z}_k\right] \tag{6.2}$$

which is the well known Shift-and-Add algorithm for integer pixel motions [63]. Plugging this estimate back into the cost function C_{ML} we obtain

$$\begin{aligned} C_{ML}(\mathbf{f}, \vec{\mathrm{v}})|_{\mathbf{f}=\hat{\mathbf{f}}} &= \sum_k \|\mathbf{z}_k - \mathbf{D}\mathbf{U}_k \hat{\mathbf{f}}\|_2^2 \\ &= \sum_k \left\| \mathbf{z}_k - \mathbf{D}\mathbf{U}_k \left[\sum_k \mathbf{U}_k^T \mathbf{D}^T \mathbf{D} \mathbf{U}_k\right]^{-1} \left[\sum_k \mathbf{U}_k^T \mathbf{D}^T \mathbf{z}_k\right] \right\|_2^2 \qquad (6.3) \end{aligned}$$

Future work on superresolution must address the minimization of the nonlinear estimation problem that is (6.3). Such an optimization problem is not easy to solve, but will undoubtable improve the shortcomings in performing super-resolution associated with the standard two step procedure of estimating motion followed by image reconstruction.

6.2.5 Performance Analysis of Orientation Estimation

In this thesis, we have analyzed the performance bound on the estimation of translation for a pair of images. If we make the assumption that the unknown translation is constant for a local (in space and time) region Ω in the image sequence, the problem of motion estimation becomes intimately connected to the problem of orientation estimation. To see this, we note that when the image sequence is of the form $f(x_1, x_2, t) = f(x_1 - v_{0_1}t, x_2 - v_{0_2}t, 0)$, we see that the function f is actually a 2-D function embedded in a 3-D space, consisting of parallel lines of constant gray levels. The problem of translation estimation for a local region in space-time becomes that of estimating the orientation of these parallel lines [83].

Without loss of generality, we present the lower dimensional problem of estimating orientation in a 2-D plane. For such a scenario, we assume that locally the image function is given by

$$z(x_1, x_2) = \eta(\mathbf{x}^T\mathbf{n}) + \epsilon(x_1, x_2), \ \mathbf{x} \in \Omega \tag{6.4}$$

where $\mathbf{x} = [x_1, \, x_2]^T$ and $\mathbf{n} = [\cos\varphi, \, \sin\varphi]^T$ denotes the unit length orientation vector. This model finds use in many image processing applications where it is of interest to find the dominant directional orientation $\mathbf{n}$ of the texture present in images. This problem is very similar to the problem of array processing in the signal processing literature [84]. One fundamental difference between the two problems is that in array processing, the function η is typically assumed to be a narrowband signal which significantly simplifies the problem.

In the image processing domain, the signal is no longer narrowband and the goal is to estimate the orientation vector field $\mathbf{n}(x_1, x_2)$ for an entire image. Applications using such vector fields have ranged from biometrics such as fingerprint similarity measures [85] to the design of directional filters for image data [86]. The local orientation can be thought of as the vector $\mathbf{n}$ which is perpendicular to the gradient field $\nabla z(x_1, x_2)$ on average over some local region Ω. The problem of finding such a local image orientation can be formulated as a maximization problem of the following function

$$\mathcal{C}(\mathbf{n}) = \sum_{x_1,x_2 \in \Omega} \kappa(\mathbf{n}^T \nabla z) \tag{6.5}$$

subject to the constraint that $||\mathbf{n}|| = 1$. A standard choice for the cost function κ is the quadratic functional which leads to

$$\mathcal{C}(\mathbf{n}) = \sum_{x_1,x_2 \in \Omega} (\mathbf{n}^T \nabla z)^2 = \sum_{x_1,x_2 \in \Omega} \mathbf{n}^T (\nabla z (\nabla z)^T) \mathbf{n} \tag{6.6}$$

Given the constraint that $||\mathbf{n}|| = 1$, the problem as stated is a general eigenvalue problem where the solution to the optimization problem is the eigenvector corresponding to the largest eigenvalue of the matrix $\sum_{x_1,x_2 \in \Omega} \nabla z (\nabla z)^T$. This solution has been noted in the past [83, 87, 88].

The solution is the eigenvector or basis vector which best represents the collection of gradient vectors. This problem is an example of the canonical problem of finding an optimal representation of a vector field. Currently, this process is applied locally to a collection of image regions to approximate the spatially varying orientation vector field $\mathbf{n}(x_1, x_2)$. Unfortunately, this approach fails to consider the underlying topological and geometric structure of the vector field. For instance, the orientation vector field must satisfy the global property of being curl-free. It would be interesting to study the performance limits in estimating the orientation vector field with additional information relating to the global topology of the orientation vector field. For instance, in computer graphics, it is well known that a sufficiently smooth vector field can be decomposed using the Helmholtz-Hodge decomposition [89]. Such a decomposition distinguishes the curl-free, divergence-free, and the harmonic components of a vector field. The divergence and curl free components are uniquely identified by the location of the sources and sinks and vortices respectively. It would be interesting to study the performance bounds in detecting and localizing these components. Armed with such knowledge, one might explore novel methods for finding local, statistically optimal orientation estimates which are coupled across space in novel ways to integrate such global information. Finally, one might study the statistical properties of such local estimators to find more robust versions of (6.5) using a cost function other than quadratic. Specifically, one can attempt to create a robust solution using a technique similar in spirit to the bilateral filter [90].

Bibliography

[1] O. Nestares and D. Heeger, “Robust multiresolution alignment of MRI brain volumes,” *Magnetic Resonance in Medicine*, vol. 43, pp. 705–715, 2000.

[2] B. K. Horn, *Robot Vision*. Cambridge: MIT Press, 1986.

[3] C. Cedras and M. Shah, “Motion based recognition: A survey,” *IEEE Proceedings Image and Vision Computing*, vol. 13, pp. 129–155, March 1995.

[4] M. Zucchelli, J. Santos-Victor, and H. I. Christensen, “Constrained structure and motion estimation from optical flow,” *16th International Conference on Pattern Recognition*, vol. 1, August 2002.

[5] J. Davis and S. Taylor, “Analysis and recognition of walking movements,” *16th International Conference on Pattern Recognition*, vol. 1, August 2002.

[6] M. Nicolescu and G. Medioni, “Motion segmentation with accurate boundaries - a tensor voting approach,” *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[7] M. Irani and P. Anandan, “A unified approach to moving object detection in 2d and 3d scenes,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, June 1998.

[8] J. Cohn, A. Zlochower, J.-J. J. Lien, and T. Kanade, “Feature-point tracking by optical flow discriminates subtle differences in facial expression,” pp. 396 – 401, April 1998.

[9] T. Tian, C. Tomasi, and D. Heeger, “Comparison of approaches to egomotion computation,” *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 315–320, June 1996.

[10] F. Defaux and J. Konrad, “Robust, efficient and fast global motion estimation for video coding,” *IEEE Transactions on Image Processing*, vol. 9, pp. 497–501, March 2000.

[11] Y. Keller and A. Averbuch, “Fast gradient methods based on global motion estimation for video compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 300–309, April 2003.

[12] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, 2002.

[13] H. Liu, T.-H. Hong, M. Herman, T. Camus, and R. Chellappa, “Accuracy vs. efficiency trade-offs in optical flow algorithms,” *Computer Vision and Image Understanding*, vol. 72, pp. 271–286, December 1998.

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

[15] K. R. Castleman, *Digital Image Processing*. Toronto: Prentice Hall, 1996.

[16] H.-M. Hang, Y.-M. Chou, and S.-C. Cheng, “Motion estimation for video coding standards,” *Journal of VLSI Signal Processing - Systems for Signal, Image, and Video Technology*, pp. 113–136, November 1997.

[17] A. Maintz and M. Viergever, “A survey of medical image registration,” *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, 1998.

[18] G. K. Rohde, A. Aldroubi, and B. M. Dawant, “The adaptive bases algorithm for intensity-based nonrigid image registration,” *IEEE Transactions on Medical Imaging*, pp. 1470–1479, November 2003.

[19] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, “Fast and robust multi-frame superresolution,” *IEEE Transactions on Image Processing*, vol. 13, October 2004.

[20] P. Milanfar, “Projection-based, frequency-domain estimation of superimposed translational motions,” *Journal of the Optical Society of America*, vol. 13, pp. 2151–2162, November 1996.

[21] D. Robinson and P. Milanfar, “Accuracy and efficiency tradeoffs in using projections for motion estimation,” *Proceedings of the 35th Asilomar Conference on Signals, Systems, and Computers*, November 2001.

[22] C. Tu, T. D. Tran, J. L. Prince, and P. Topiwala, “Projection-based block matching motion estimation,” *Proc. SPIE Applications of Digital Image Processing XXIII*, pp. 374–384, August 2000.

[23] S. R. Deans, *The Radon Transform and Some of Its Applications*. New York: John Wiley and Sons, Inc., 1983.

[24] S. Alliney and C. Morandi, “Digital image registration using projections,” *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 222–233, March 1986.

[25] S. Rajala, A. Riddle, and W. Snyder, “Application of the one-dimensional Fourier transform for tracking moving objects in noisy environments,” *Computer Vision, Graphics, and Image Processing*, vol. 21, pp. 280–293, 1983.

[26] J.-S. Kim and R.-H. Park, “A fast feature-based block matching algorithm using integral projections,” *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 968–971, June 1992.

[27] S. C. Cain, M. M. Hayat, and E. E. Armstrong, “Projection-based image registration in the presence of fixed-pattern noise,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1860–1872, December 2001.

[28] F. Coudert, J. Benois-Pineau, and D. Barba, “Dominant motion estimation and video partitioning with a 1-D signal approach,” *SPIE Conference on Multimedia Storage and Archiving Systems III*, vol. 3527, pp. 283–294, 1998.

[29] A. Akutsu and Y. Tonomura, “Video tomography: An efficient method for camerawork extraction and motion analysis,” *Transactions of the Institute of Electronics, Information, and Communications Engineers*, vol. J79D-II, pp. 675–686, May 1996.

[30] S. A. Seyedin, “Motion estimation using the Radon transform in dynamic scenes,” *Proceedings of the International Society for Optical Engineering*, vol. 2501, pp. 1337–1348, 1995.

[31] T. Tsuboi, A. Masubuchi, and S. Hirai, “Video-frame rate detection of position and orientation of planar motion objects using one-sided Radon transform,” *Proceedings IEEE Conference of Robotics and Automation*, vol. 2, pp. 1233–1238, April 2001.

[32] J. You, W. Lu, J. Li, G. Gindi, and Z. Liang, “Image matching for translation, rotation, and uniform scaling by the Radon transform,” *Proceedings International Conference on Image Processing*, vol. 1, pp. 847–851, 1998.

[33] P. Milanfar, “A model of the effect of image motion in the Radon transform domain,” *IEEE Transactions on Image Processing*, vol. 8, pp. 1276–1281, September 1999.

[34] D. Robinson and P. Milanfar, “Fast local and global projection- based methods for affine motion estimation,” *Journal of Mathematical Imaging and Vision*, vol. 18, pp. 35–54, January 2003.

[35] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimationg Theory*. Prentice Hall Inc., 1993.

[36] C. Bergeron and E. Dubois, “Gradient-based algorithms for block-oriented MAP estimation of motion and application to motion-compensated temporal interpolation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp. 72–85, March 1991.

[37] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani, “Hierachical model-based motion estimation,” *Proceedings European Conference on Computer Vision*, pp. 237–252, 1992.

[38] M. J. Black and P. Anandan, “The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields,” *Computer Vision and Image Understanding*, vol. 63, pp. 75–104, January 1996.

[39] A. Jepson and M. Black, “Mixture models for optical flow computation,” *Proceedings Computer Vision and Pattern Recognition*, pp. 760–761, June 1993.

[40] C. Stiller and J. Konrad, “Estimating motion in image sequences,” *IEEE Signal Processing Magazine*, vol. 16, pp. 70–91, July 1999.

[41] M. T. Heath, *Scientific Computing: An introductory survey*. New York: McGraw-Hill, 2002.

[42] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *DARPA81*, pp. 121–130, 1981.

[43] H. S. Stone, M. Orchard, and E.-C. Chang, “Subpixel registration of images,” *Proceedings of the 1999 Asilomar Conference on Signals, Systems, and Computers*, October 1999.

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

[45] Q. Tian and M. Huhns, “Algorithms for subpixel registration,” *Computer Vision, Graphics, and Image Processing*, vol. 35, pp. 220–233, 1986.

[46] G. Jacovitti and G. Scarano, “Discrete time techniques for time delay estimation,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 525–533, February 1993.

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

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

[49] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: Wiley, 1968.

[50] A. Nehorai and M. Hawkes, “Performance bounds for estimating vector systems,” *IEEE Transactions on Signal Processing*, vol. 48, pp. 1737–1749, June 2000.

[51] J. Shi and C. Tomasi, “Good features to track,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, June 1994.

[52] D. J. Field, “Relations between the statistics of natural images and the response properties of cortical cells,” *Journal of the Optical Society of America A*, vol. 4, pp. 2379–2393, December 1987.

[53] S. Dooley and A. Nandi, “Comparison of discrete subsample time delay estimation methods applied to narrowband signals,” *IOP Measurement and Science Technology*, vol. 9, pp. 1400–1408, September 1998.

[54] R. Moddemeijer, “On the determination of the position of extremum of sample correlators,” *IEEE Transactions on Signal Processing*, vol. 39, pp. 216–219, January 1991.

[55] Q. C. Davis and D. M. Freeman, “Statistics of subpixel registration algorithms based on spatiotemporal gradients or block matching,” *Optical Engineering*, pp. 1290–1298, April 1998.

[56] V. Dvornychenko, “Bounds on (determinisitic) correlation functions with applications to registration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 206–213, March 1983.

[57] J. Kearney, W. Thompson, and D. Boley, “Optical flow estimation: an error analysis of gradient-based methods with local optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 229–244, March 1987.

[58] J. W. Brandt, “Analysis of bias in gradient-based optical flow estimation,” *IEEE Asilomar Conference on Signals, Systems and Computers*, pp. 721–725, 1995.

[59] C. Fermuller, D. Shulman, and Y. Aloimonos, “The statistics of optical flow,” *AP Computer Vision and Image Understanding*, vol. 82, pp. 1–32, 2001.

[60] H.-H. Nagel and M. Haag, “Bias-corrected optical flow estimation for road vehicle tracking,” *Proceedings of the International Conference on Computer Vision*, pp. 1006–1011, 1998.

[61] M. Elad, P. Teo, and Y. Hel-Or, “On the design of optimal filters for gradient-based motion estimation,” *International Journal on Mathematical Imaging and Vision submitted*, 2003.

[62] E. Simoncelli, “Design of multi-dimensional derivative filters,” *Proceedings IEEE ICIP 1994*, 1994.

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

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

[65] P. Vandewalle, L. Sbaiz, S. Susstrunk, and M. Vetterli, “How to take advantage of aliasing in bandlimited signals,” *Proc. IEEE Conference on Acoustics, Speech and Signal Processing*, pp. 948–951, May 2004.

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

[67] F. Graybill, *Matrices with Applications in Statistics*. Belmont: Wadsworth Publishing Company, 1969.

[68] L. Scharf and T. McWhorter, "Geometry of the Cramér-Rao bound," *Signal Processing*, vol. 31, pp. 303–311, 1993.

[69] P. Stoica and T. Marzetta, "Parameter estimation problems with singular information matrices," *IEEE Transactions on Signal Processing*, vol. 49, pp. 87–90, January 2001.

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

[71] M. Unser and J. Zerubia, "Generalized sampling: Stability and performance analysis," *IEEE Transactions on Signal Processing*, vol. 45, pp. 2941–2950, December 1997.

[72] M. Elad and Y. Hel-Or, "A fast super-resolution reconstruction algorithm for pure translational motion and common space invariant blur," *IEEE Transactions on Image Processing*, vol. 10, pp. 1186–1193, August 2001.

[73] D. Harville, *Matrix Algebra: Exercises and Solutions*. Springer, 2001.

[74] R. Bracewell, *The Fourier Transform and its Applications*. New York: McGraw-Hill, 1999.

[75] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 24, pp. 1167–1183, September 2002.

[76] D. Robinson and P. Milanfar, "Fundamental performance limits in image registration," *IEEE Transactions on Image Processing*, September.

[77] D. Robinson and P. Milanfar, "Bias-minimizing filters for motion estimation," *Proceedings of the 37th Asilomar Conference on Signals, Systems, and Computers*, November 2003.

[78] J. Ziv and M. Zakai, "Some lower bounds on signal parameter estimation," *IEEE Transactions on Information Theory*, pp. 386–391, May 1969.

[79] K. L. Bell, Y. Steinberg, Y. Ephraim, and H. V. Trees, "Extended Ziv Zakai lower bound for vector parameter estimation," *IEEE Transactions on Information Theory*, vol. 43, pp. 624–637, March 1997.

[80] E. Barankin, "Locally best unbiased estimates," *Annals of Mathematical Statistics*, no. 20, 1949.

[81] A. Zeira and P. Schultheiss, “Realizable lower bounds for parameter estimation,” *IEEE Transactions on Signal Processing*, vol. 41, pp. 3102–3113, November 1993.

[82] A. Hero, J. Fessler, and M. Usman, “Exploring estimator bias variance tradeoffs using the uniform CR bound,” *IEEE Transactions on Signal Processing*, vol. 44, pp. 2026–2024, August 1996.

[83] J. Bigün, G. H. Granlund, and J. Wiklund, “Multidimensional orientation estimation with applications to texture analysis and optical flow,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 775–790, August 1991.

[84] H. Krim and M. Viberg, “Two decades of array signal processing research,” *IEEE Signal Processing Magazine*, pp. 67–94, July 1996.

[85] N. Ratha, S. Chen, and A. Jain, “Adaptive flow orientation-based feature extraction in fingerprint images,” *Pattern Recognition*, vol. 28, pp. 1657–1672, 1995.

[86] J. Starck, E. Candes, and D. Donoho, “The curvelet transform for image denoising,” *IEEE Transactions on Image Processing*, vol. 11, pp. 670–684, 2002.

[87] P. Perona, “Orientation diffusions,” *IEEE Transactions on Image Procssing*, vol. 7, pp. 457–467, 1998.

[88] G. Farneback, “Orientation estimation based on weighted projections onto quadratic polynomials,” *Vision, Modeling and Visualization 2000*, pp. 89–96, 2000.

[89] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun, “Discrete multiscale vector field decomposition,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 445–452, 2003.

[90] M. Elad, “On the origin of the bilateral filter and ways to improve it,” *IEEE Transactions on Image Processing*, pp. 1141–1151, October 2002.