Subpixel Motion Estimation From Sequences of Video Images

by

Samson J. Timoner

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1999
[February 2000]

Author ....
Department of Electrical Engineering and Computer Science
June 11, 1999

Certified by ....
Dennis M. Freeman
Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Subpixel Motion Estimation From Sequences of Video Images

by

Samson J. Timoner

Submitted to the Department of Electrical Engineering and Computer Science
on June 11, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

We develop a new class of multi-image gradient-based algorithms, and a discrete Fourier transform based algorithm to detect subpixel motions of objects in video images. Because of their enormous practical importance, we make estimators of amplitude and phase of temporal sinusoidal motion using both methods.

We show that to improve motion estimates of existing gradient-based algorithms, it is not sufficient to improve spatial gradient estimates alone; it is necessary to improve both spatial and temporal gradient estimates. We use data in many images to estimate spatial and temporal derivatives to high accuracy. By using many images, we are also able to compensate for the blur caused by the finite image acquisition times.

Through analysis of simple images and through simulations, we show that the inherent bias of multi-image gradient-based methods can be made arbitrarily small for small motions. However, for large motions, multi-image gradient based methods cease to perform well. We simulate the performance of our algorithms in the presence of noise typical of optical microscopes and scientific grade cameras. These simulations show that the sinusoidal estimators we create achieve errors below 0.001 pixels and 0.001 radians for amplitudes smaller than 1.2 pixels. However, for motions larger than 2 pixels, the amplitude errors are larger than 0.1 pixels.

We show that Fourier transform based methods are limited by bias inherent to the method for the analysis region sizes that interest us. In the presence of noise typical for optical microscopes, the sinusoidal estimator we create achieves 0.1 pixel accuracy and 0.01 radian accuracy. These inaccuracies are greater than those of already existing algorithms.

We experimentally test the performance of the multi-image gradient-based sinusoidal estimators. We show that the algorithms achieve nanometer accuracy and precision for motions below 500 nanometers. The results agree well with the predicted performance of the algorithm based on simulations. We also show that the algorithms are consistent to within a nanometer across regions of the same moving object with very different brightness These features of the new algorithms represent important improvements over existing algorithms.

Thesis Supervisor: Dennis M. Freeman
Title: Associate Professor of Electrical Engineering

Acknowledgments

Dr. C. Quentin Davis was invaluable by providing comments and by participating in several insightful conversations. Special thanks to Professor John Wyatt who helped me remain excited about this project. He also deserves credit for introducing me to Trefethen's book [37] which gave me important insights into some of the problems I faced. Professor Dennis M. Freeman deserves recognition for reasons too numerous to list here. Denny deserves special credit for teaching me how to write technically.

Proof readers helped me develop this work from a disorganized, unintelligible grouping of ideas into an understandable document. I owe a great deal of thanks to Dr. Werner Hemmert, Jekwan Ryu, Michael Gordon, Dr. C. Quentin Davis, and of course Professor Dennis. M. Freeman.

I am grateful for the support of the Fannie and John Hertz Organization who pays the great majority of my costs as a student. This work was also supported by a grant from DARPA (F30602-97-2-0106).

Finally, I recognize some of the outstanding work in the literature. While H.H. Nagel has produced several overly-long papers, a few of them are brilliant and essential to the understanding of gradient-based methods. Ng and Solo's [33] introduction of Sprent's work [35] into the motion estimation field is an essential ingredient to the noise analysis of gradient-based methods. Davis and Freeman's work [7, 8, 9, 10] is crucial to the subpixel estimation field, but not because the work is so revolutionary. Davis and Freeman deserve respect because they had the audacity to think their methods would work. Finally, Horn's initial work in the field was instrumental in the development of optical flow and gradient-based methods.

In memory of

Rabbi and U.S. Army Captain Chana Timoner

August 24, 1951 - July 13, 1998

Contents

Chapter 1

Introduction

The exponential growth and wide-spread availability of inexpensive computer memory, disk storage and fast computers is radically increasing our ability to process information from optical imaging systems. We can now analyze gigabytes of data from scientific grade CCD cameras quickly and inexpensively. The ability to process huge quantities of data in conjunction with inexpensive video imagers is creating new applications for video-systems that were not previously practical.

Video microscopy holds promise to revolutionize our ability to make quantitative measurements. We explore methods to use video images to estimate nanometer motions. In this thesis, we investigate ways to harness the growing computational power of computers to increase the accuracy of subpixel motion estimators.

1.1 Background and Motivation

Recently, Davis and Freeman developed algorithms that can measure three-dimensional motions between two volume images to hundredths of the distance between voxels [7]. Surprisingly, it is bias inherent to their algorithm, not noise, that limits performance. We seek to develop an estimator of motion whose performance is not limited by errors inherent to the algorithm.

Several classes of algorithms are widely used to determine motions from video images including gradient based methods, correlation based methods and transform based methods. We investigate these classes with the goal of understanding which classes of algorithms have the possibilities of yielding low bias estimates.

We note that many motion estimation techniques register two images using information from only those two volume images. In many circumstances, sequences of images exist before and after the two images of interest. We extend gradient-based methods to use data from a sequence of images to estimate the motion between two images. By using information from additional images, better estimates of spatial and temporal gradients are found, leading to significant bias reductions in the estimate.

A review of the motion estimation techniques that have previously been reported suggests another promising technique, Fourier transform based methods. These methods use the phase difference between the Fourier transform of two images to detect motion. We pursue techniques similar to already existing methods [25, 1].

We are specifically interested in examining the motions of bio-mechanical and microelectromechanical systems (MEMS). For these systems, sinusoidal motion is of great practical importance because one often wants to examine the modes of the system. Therefore, we develop estimators of amplitude and phase of temporal sinusoidal motion.

1.2 Document Summary

In Chapter 2, we review video imaging in conjunction with optical microscopy. We examine data collection techniques, the properties of the resulting signals, and the noises introduced. The key conclusion of this chapter is that above spatial frequencies 2, the signal to noise ratio in the images is small.

We review the different classes of algorithms that exist in the motion estimation literature in Chapter 3. We discuss each method and explore the applicability to the challenges addressed in this thesis. We conclude that for rigid-body translations, many techniques are not worth pursuing. We decide to pursue gradient-based methods, and Fourier transform based methods as these methods have the potential to measure very small subpixel motions.

In Chapter 4, we examine the limitations of gradient-based methods. Specifically, we show that gradient-based methods yield estimates of velocity, not displacement. We also show that these methods make large errors for large motions. The key conclusion of this chapter is that the accuracy of a gradient-based algorithm is limited by the derivative estimated with the largest error. Therefore, to make a low bias gradient-based algorithm, it is not sufficient to calculate high accuracy spatial gradients alone. It is necessary to estimate high-accuracy spatial and temporal derivatives.

In Chapter 5, we illustrate the design of multi-image gradient-based methods. We use the methods to develop an estimator of amplitude and phase of temporal sinusoidal motion. Simulations in the presence of noise typical of optical systems show that amplitude estimates with 0.001 precision and accuracy and phase estimates with 0.001 radian precision and accuracy are achievable.

We develop a Fourier transform based estimator of temporal sinusoidal motion in Chapter 6. Simulations in the presence of noise show that amplitude errors of 0.1 pixels and phase errors of 0.01 radians are achievable. However, these errors are larger than those of already existing algorithms. Therefore, we do not experimentally test these algorithms.

In Chapter 7 we verify the simulations of the multi-image gradient-based methods in the presence of noise by experimentally testing the algorithms. Using video microscopy, we examine the motions of a bead and a MEMS device attached to a piezo. Generally, the simulations match the experimental findings. The algorithms achieve 1 nanometer accuracy and precision for motions smaller than 500 nanometers.

Chapter 2

Properties of the Data

This thesis is primarily concerned with the development of subpixel motion estimators. To develop a good estimator, we examine properties of the data. In this chapter we briefly review the data acquisition process, the resulting signals, and the noises introduced.

2.1 Data Acquisition

A light microscope, in combination with stop-action photography and optical sectioning techniques, collects three-dimensional images of mechanical structures whose motions are to be examined. Freeman and colleagues dubbed the data collection process "computer microvision" [13]. We briefly review computer microvision emphasizing how the data collection process affects the data. A more detailed account of the method is found in [13].

2.1.1 Three-Dimensional Video Microscopy

Mechanical structures whose motions are to be measured are affixed to the base of an optical microscope which sits on an vibration isolation table. The lens system in the microscope magnifies the image of the mechanical structure and projects the image onto a CCD camera. To collect three-dimensional data, a series of two-dimensional images are acquired at evenly spaced focal planes as shown in Figure 2-1. The focal plane is adjusted by applying a voltage to a piezoelectric crystal that translates a lens along the optical axis.

2.1.2 Stroboscopic Illumination

A sinusoidal stimulus, typically of kilohertz frequencies, excites the sample. Since most scientific grade cameras collect images at the rate of tens of hertz, a light emitting diode (LED) strobes the sample to image faster motion.

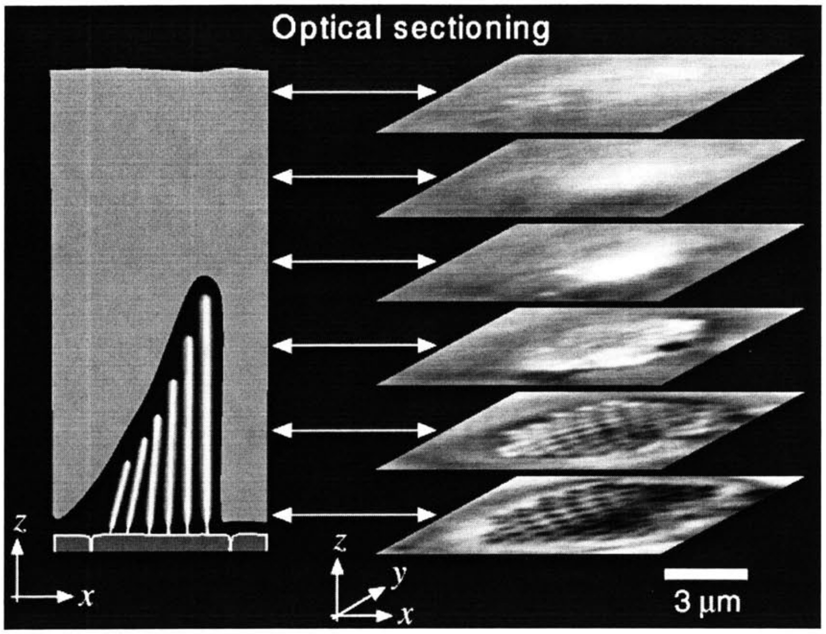

Figure 2-1: Illustration of optical sectioning. A three-dimensional volume image is generated by capturing a sequence of two-dimensional images at different focal planes. The focal plane is adjusted by application of a voltage to a piezo that translate the lens along the optical axis.

An LED must flash many times in order to provide sufficient light to use the full dynamic range of the camera. To obtain one plane of the three-dimensional image, the LED flashes at the same phase in the stimulus cycle many times.

Repeated application of stroboscopic illumination at different image planes and at different phases generates a four-dimensional image (three spatial and one temporal dimension) of the moving sample. This four-dimensional image is the data set from which we seeks to determine motion.

2.2 Signal and Noises

We examine the effects of the data acquisition process on the resulting images. We then consider the noises introduced during the data collection. We divide the noise sources into two categories. The first source of noise is due to errors in image collection. The second source of noise is mechanical.

2.2.1 Properties of the signals

The lens system that collects the images can be modeled as a spatially and temporally invariant low-pass filter [16]. The CCD camera has pixels with non-zero length and width. Each pixel effectively averages the light signal over a small region, a process which can be modeled as sampling a low-pass filtered version of the image. If the pixel size is small enough, the CCD camera will oversample the continuous intensity profile resulting in a two-dimensional band limited image. If the focal planes are spaced closely enough along the optical axis, the three-dimensional volume image will be oversampled.

2.2.2 Image Acquisition Noises

The CCD camera introduces quantization noise, shot noise, fixed-pattern noise and read noise [21]. For a 12-bit scientific grade camera (0-4096 gray values) under peak lighting conditions, shot noise typically dominates quantization noise and read noise.

Fixed-pattern noise results from the varying gain of pixels in the CCD camera and from dirt on lenses. Dirt causes variations in illumination which do not change when the object translates. See Figure 2-2 for an example. Davis and Freeman describe a process of two-point correction [8] that decreases fixed-pattern noise by 5 to 10 dB.

Related to fixed-pattern noise are confounding notions. Many samples have openings in them through which stationary background patterns are visible. If an algorithm cannot detect these non-moving regions, the stationary patterns can create effects similar to those caused by dirt on the optics.

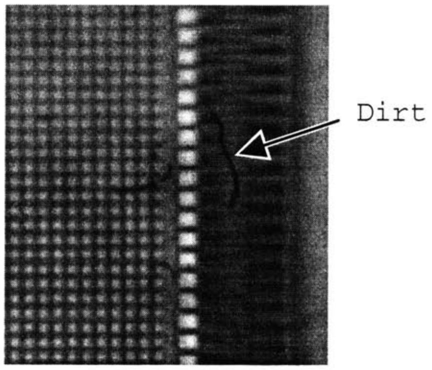

Figure 2-2: A example of fixed-pattern noise due to dirt on a lens. The image is one in a sequence of images of part of a moving gyroscope (Draper Labs, Cambridge MA). The dirt, indicated by the arrow, was observed to remain stationary while the gyroscope moved underneath it.

An additional source of error in the image acquisition process occurs due to the finite LED intensity. To obtain sufficient light, the LED illuminates the sample for a fraction of the stimulation cycle, resulting in blurring of moving objects in the acquired image. This process can be modeled as a low-pass filter in time.

An additional error in the data collection is due to nonlinearities in the lens. While a lens can be approximated as a spatially invariant linear low-pass filter; it is not perfectly linear nor perfectly spatially invariant. It is not clear how large or how important these effects are.

2.2.3 Mechanical Noise

Relative motion between the CCD camera, the optical system and the sample will be interpreted by an algorithm as additional motion of the sample. Therefore mechanical vibrations in the data collection system lead directly to motion estimation errors.

A typical vibration isolation table has a resonance near a few hertz. Images collected at slower rates are particularly susceptible to the low frequencies which pass through the table. Since the CCD camera collects each plane in the image at a different time, vibrations result in net displacements and rotations between planes of the same volume image, as well as rotations and displacements between different volume images.

The LED flashes many times to collect one image plane. The table will move during the collection of one plane. Thus the collection of one frame can be modeled as an averaging process, with the pixel averaging a volume of space as it is jiggled. For long enough averaging times, we expect the vibrations of the table to be averaged out. Davis and Freeman found this to be the case [8].

Note that the microscope and the sample may have significant displacements at very low frequencies due to, for example, thermal expansion. Over the course of an experiment, an object may "drift" in one direction which will also introduce error into the motion estimates.

2.2.4 Conclusions

The image acquisition process is limited by shot noise, which introduces variability into the images, and fixed-pattern noise, which does not move when an object in an image does. Additionally, relative mechanical vibrations between the camera and the optics will introduce errors into the motion estimates.

2.3 Spectral Properties of the Data

In this section we characterize acquired data by examining the power spectrum of several images. We seek to determine the relative powers in the shot noise and fixed-pattern noise, as well as how the signal to noise ratio varies as a function of spatial frequency. The data in this section will be used to design motion estimators in later chapters.

2.3.1 Methods

Images of specimens were collected using a light microscope. The images were two-point corrected and then analyzed to determine their power spectra and their noise power spectra.

Images

We examine three images. The first is a 256 by 256 pixel image of a 1 μm polystyrene bead (Lot #46747, Polysciences, Inc, Warrington PA) dried onto a glass cover slip. This image was chosen as an example with a small ratio of signal to noise. The second is a 512x512 image of a polysilicon fatigue test structure designed by Exponent Failure Analysis Associates. We refer to the device as the fatigue device. This image reflects our interest in measuring the motions of silicon based micromechanical structures. The third image is a 256 by 256 pixel picture of an alligator lizard cochlea. This image reflects our interests in biological motion estimation applications. All three images have approximately the same average intensity.

Image Collection

The one micrometer bead was imaged with a light microscope (Zeiss, Axioplan, Thorwood, NY) using transmitted light, a long working distance condenser with 0.6 numerical aperture (0.6 NA) and a 20x, 0.4 NA objective with a 1x optivar. Images were captured using a 12-bit CCD camera with 12 μm square pixels (DALSA CA-D7, Waterloo, Ontario) with an exposure time of 80 ms. The fatigue device was imaged using the same setup, except using epi-illumination with an exposure time of 790 ms.

The alligator lizard cochlea was collected with the same light microscope using transmitted light with a 0.75 NA, 40x water-immersion lens with a 2.5 optivar for a total magnification of 100x. The condenser NA was 0.6 so that the effective NA of the system was closer to 0.6. The CCD camera was a Photometrics CH250A CCD camera with 23 μm spacing, almost twice the spacing of the DALSA camera used in the previous two examples.

Two-Point Correction

To reduce the effect of fixed-pattern noise, we use a two-point correction technique. First we address fixed-pattern offsets in the absence of light. After a microscope is set up to image a specimen, many images are acquired with no sample illumination. Between 50 and 128 of these images are averaged to create a "dark" image. To address the varying sensitivities of individual pixels, many images of uniformly bright regions are averaged together. To obtain a "bright" image for the bead and alligator lizard we focused on the specimen and then moved the specimen to the side so that the light path from the condenser to the imaging lens passed through air only. To obtain bright images for the fatigue device, we replaced the fatigue device with a reflecting silicon wafer. Between 50 and 128 images were averaged to obtain one bright image.

Images are corrected for pixel-to-pixel variations in sensitivity and offset by calculating a corrected image

$$G[i,j] = \frac{M[i,j] - D[i,j]}{B[i,j] - D[i,j]}$$

where $M[i,j]$ represents the gray value generated by the $[i,j]^{\text{th}}$ pixel in the measured image, and $B[i,j]$ and $D[i,j]$ represent corresponding gray values in the bright and dark images, respectively.

Power Spectrum Calculations

In all cases, "spectral density" refers to the squared magnitude of the two-dimensional Fourier transform of an image that was first multiplied by a two-dimensional Hanning window. The spectral density is scaled by the number of pixels in the image. If there were no window function, the D.C. value of the spectral density would therefore be the average signal in the image squared in units of gray values squared. We present the cross-section of the two-dimensional Fourier transform along one axis of the CCD camera. All images are shown so that the cross-section is parallel to the bottom of the images. In most cases the D.C. value is several orders of magnitude larger than other values and is therefore removed from the plot so that details in the rest of the spectrum are more evident.

For the bead and polysilicion images, the power spectrums were obtained by averaging the power spectrums of 32 different, two-point corrected images. For the alligator lizard cochlea, only one image was used.

Noise Analysis

We attempt to determine the power spectrum of the shot noise and fixed-pattern noises. As an estimate of the power spectrum of the first, we subtract the average of 32 images from each image and then average the power spectra of the residual images. The resulting power spectrum is therefore a measure of the variability in the sequence of images. To estimate fixed-pattern noise, we average the spectra of 32 two-point corrected images of a 256x256 pixel region that appear close to uniformly bright, containing little structure. The region was located several hundred pixels away from the bead.

2.3.2 One Micrometer Bead

Figure 2-3 shows the spectrum of the two-point corrected image of a one micrometer bead. Relative to the magnitude of its signal, the power spectrum of the bead has large variability from one frequency to an adjacent frequency above spatial frequency 2, with π being the Nyquist frequency. We found that two-point correction reduced the variability in the power spectrum by more than a factor of two.

We estimated the variability of images of the bead by examining the residual images found by subtracting a mean image from each initial image of the bead as described in Section 2.3.1. The average power spectrum of the residual images is shown in Figure 2-3. We examined the variability

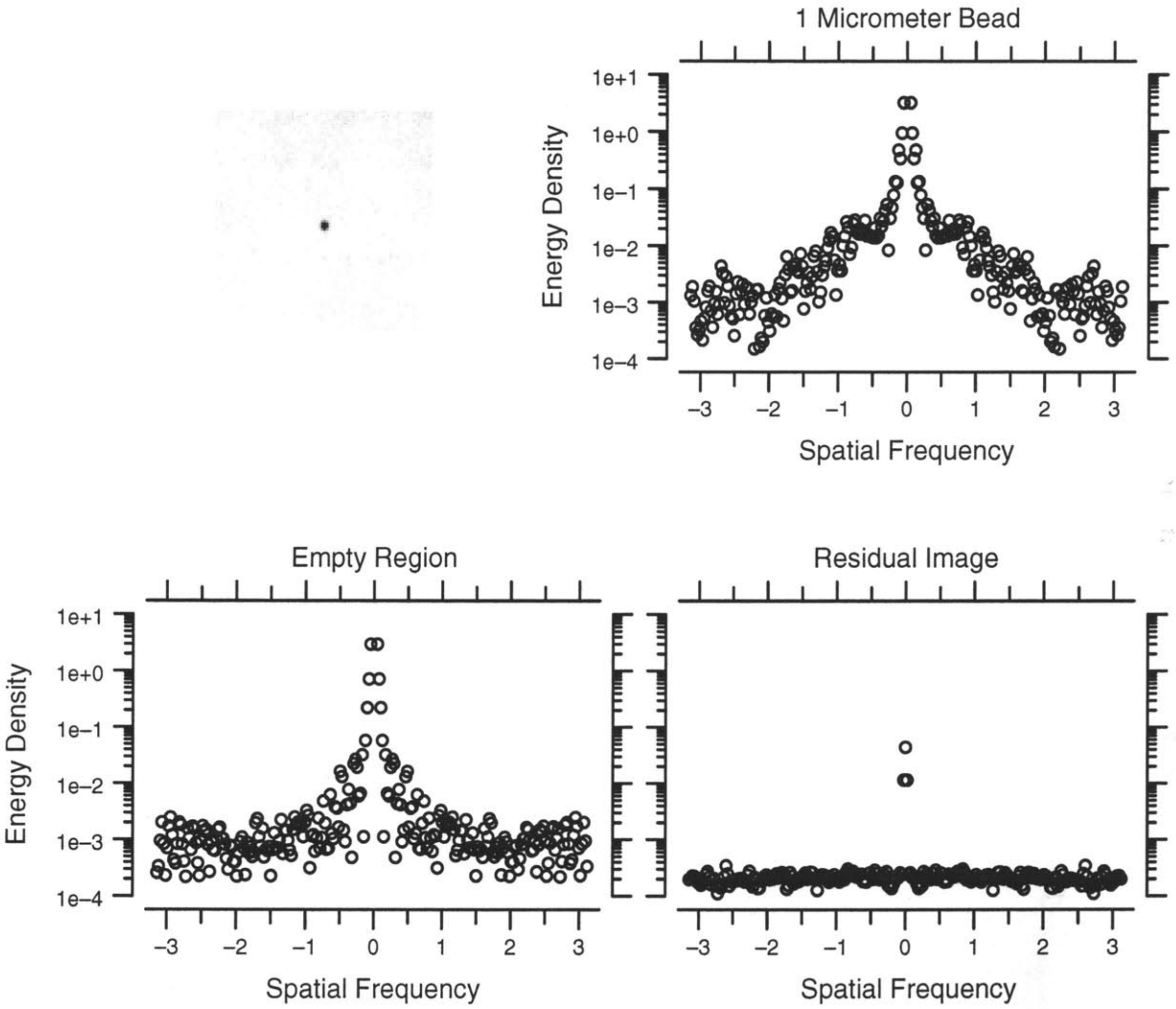

Figure 2-3: Top left: two-point corrected, 256x256 pixel image of a 1 μm diameter bead. Top right: average power spectrum of the bead image after two-point correction. Bottom left: average power spectrum of 256x256 region containing little visible structure after two-point correction. The region was located several hundred pixels away from the bead. Bottom right: average power spectrum of residual images. These images are the results of subtracting the average image of the bead from each image of the bead as described in Section 2.3.1.

of the residual images finding the standard deviation to be 9.4 gray values, or -51 dB relative to the energy in the bright transmission field. Using Dalsa's claim of a gain of 73 electrons per gray value, shot noise would account for 7 gray value units of the variability.

Also in Figure 2-3 is the power spectrum of a two-point corrected region containing little visible structure; we refer to it as an empty region. This power spectrum has approximately the same frequency to frequency variability seen in the power spectrum of the bead image. We found the standard deviation of the region before and after two-point correction to be 53 and 34 gray value units respectively, corresponding to -36.2 and -41 dB relative to the energy in the bright region. Comparing the power spectrum of the empty region and the power spectrum of the residual image, we see that there is roughly a factor of 10 more energy in the empty region.

We also examined several empty regions with dimensions between 50 and 100 pixels on a side, rather than 256 pixels. We found that the standard deviation of theses smaller regions before and after two-point correction averaged to -44 dB and and -50 dB relative to the energy in the bright region.

2.3.3 Polysilicon Fatigue Device

Figure 2-4 shows an image of the polysilicon fatigue device and its average power spectrum. The peaks in the spectrum correspond to the periodicities in the device. Note that the shape of the spectrum is qualitatively similar to the the spectrum of the bead. Specifically, for spatial frequencies larger than 2, the variability in the spectrum from one frequency to an adjacent frequency is larger than the average magnitude of the spectrum.

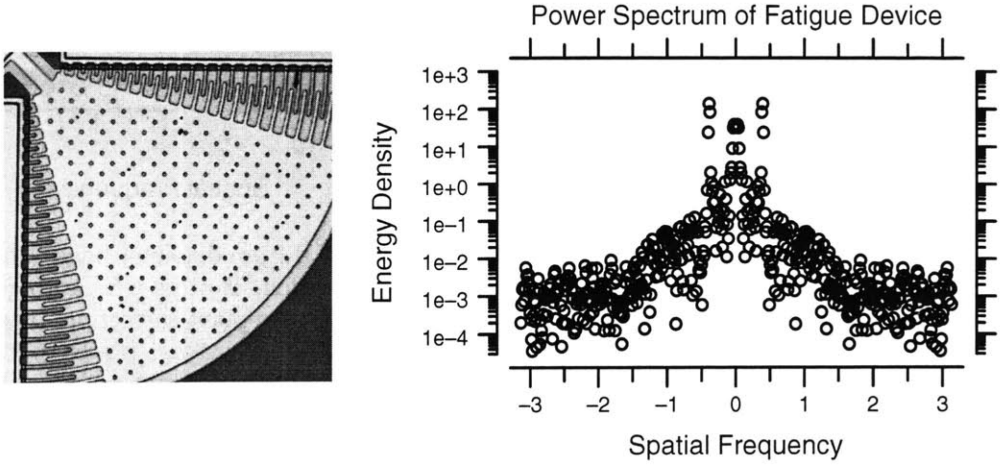

Figure 2-4: Polysilicon Failure device and its power spectrum.

We examined the residual images as we did for the bead. The average power spectrum of the residual images (not show) is flat, with the exception of several peaks that correspond to the periodicities in the device. The standard deviation of the residual images was 9.5 grey units, very close to the value found for residual images in the last section.

Unlike the image of the bead, there is no region near the fatigue device that contained no structure. We instead examined the standard deviation of the 512x512 bright field image which we found to be 60 gray value units, 13% larger than the non two-point corrected empty region near the bead. Because we used a mirror to acquire the bright field image, the bright field image may be contaminated by dirt on the mirror. We examined several rectangular regions within the bright field image with dimensions between between 50 and 100 pixels on a side that appeared to have no dirt. We found those regions to have standard deviations between 33 and 49 gray values. However, the means of these regions differed by between 30 and 40 units.

2.3.4 Alligator Lizard Ear

Figure 2-5 shows an image of an alligator lizard cochlea. The CCD that collected the image used 23 μm pixels, almost twice the dimension of the pixels used in the previous two images. Figure 2-5 shows the power spectrum of one image of the ear. Because the power spectrum is derived from only one image, the frequency to frequency variability is not a good noise measure for this image. We present this image to point out that the power spectrum of the lizard ear is similar in shape and magnitude to the other two power spectra presented.

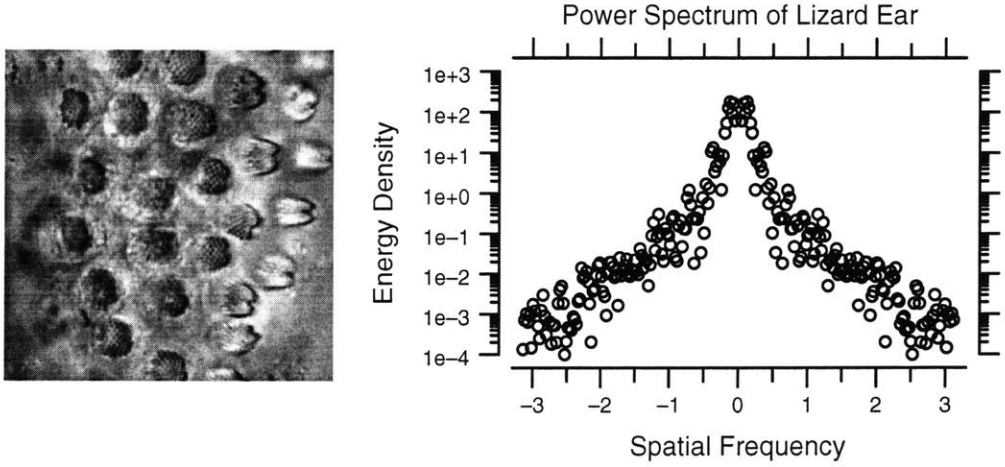

Figure 2-5: Alligator Lizard Ear and its power spectrum.

2.3.5 Discussion

We interpret the power spectrum of the residual images from the bead to be mainly due to shot noise. If the variability were entirely due to the shot noise, we would predict a gain of 37 electrons per gray value, not the 73 quoted by DALSA. In fact, using the listed saturation power, frame rate, and quantum efficiency of the camera, we calculated a gain closer to 40 electrons per gray value. That the residual images from the fatigue device gave almost the identical variability with an exposure time a factor of 10 longer bolsters the argument that the variability is due to shot noise and not, for example, dark current.

For the image of the bead, where the intensity across the image is almost uniform, one would expect the power spectrum of the shot noise to be roughly white. The uniformly bright image of the bead is particularly interesting in that the image was as bright as an experimenter would normally allow. Therefore, we expect the residual noise spectrum will always be smaller than the power spectrum of the residual images shown in Figure 2-3.

We attribute the power spectrum of the empty region near the bead as being due to fixed-pattern noise. The spectrum of this region accounts for the frequency to frequency variability of the power spectrum of the image of the bead. It also seems to account for much of the variability in the fatigue device. We note that the power spectrum of the empty region is roughly 10 dB larger than the power spectrum of the residual images. We conclude that over a 256x256 region, fixed-pattern noise is 10dB larger than shot noise.

Our measurements of empty rectangular regions with between 50 and 100 pixels on a side show almost 8 dB less fixed-pattern noise than the 256x256 empty region. That we found a similar dependence on region size in the standard deviation of the bright field images of the fatigue device suggests that fixed-pattern noise strongly depends on region size for our camera. We conclude that analysis region size should be chosen carefully for algorithms susceptible to fixed-pattern noise.

Looking at the power spectra of the bead and the fatigue device, we note that the spectra become level with large variability after spatial frequency 2. We also observe that the power spectrum of the empty region becomes levels at 3×10^{-3} energy density units, the same power at which the spectra of the images of the bead and the fatigue device become level. We conclude that the variability in the bead and fatigue device spectra is due to fixed-pattern noise and that the signal to noise ratios of the two test images are small for spatial frequencies greater than 2.

The power spectrum of the lizard ear reaches 3×10^{-3} energy density units roughly at spatial frequency 2.5. If the image were acquired on the DALSA camera, its frequency range would be condensed by a factor of two due to the greater sampling rate of the DALSA over the Photometrics camera. Thus, if the lizard image were photographed using the DALSA camera, we would expect the fixed-pattern noise would swamp the power spectrum in the lizard ear image spectrum near spatial frequency 1.75.

2.3.6 Conclusions

We examined the spectra of an image of a bead, a silicon based structure, and a biomechanical structure. We found the power spectra of all three quantitatively similar. For the bead and the fatigue device, the signal to noise ratio drops below 1 near spatial frequency 2. If the lizard ear had been acquired on the same camera, we expect the same findings.

We also examined noise spectra which we attributed to fixed-pattern noise and shot noise. We concluded that for the DALSA camera, for region sizes larger than 256x256 pixels, fixed-pattern noise is −41 dB after two-point correction, 10 dB larger than shot noise. However, we found that for region sizes between 50 and 100 pixels on a side, fixed pattern noise levels of −50 dB are achievable after two point correction.

Chapter 3

A Review of Motion Estimation Algorithms

We seek to estimate three-dimensional translations of a rigid body from a sequence of volume images. We are specifically interested in subpixel motions and attempting to measure the smallest possible detectable motions. We review motion estimation techniques in the literature with the purpose of finding estimators that are not fundamentally resolution limited.

There are several reviews and comparisons of motion detection algorithms including [36, 3, 31, 2, 28]. However, the challenges we address are very different from many of the problems faced in the motion estimation literature. Many researchers contend with varying flow fields resulting from very complicated motion. The rigid, irrotational motion assumptions allow us to ignore these complications. Most investigators don't have the luxury of three-dimensional data; many attempt to determine three-dimensional motion from two-dimensional images from one or multiple cameras. Most notably, by using scientific grade cameras in conjunction with targets whose motions are driven, we enjoy significantly larger signal to noise ratios than those of the majority of researchers. (See Chapter 2 for a description of the data acquisition process.)

In this chapter, we review many of the classes of algorithms used in motion detection. We highlight the gradient-based class of algorithms, block matching and Fourier transform methods which we pursue in later chapters. We also discuss several popular methods that are not appropriate for the challenges we address.

3.1 Gradient Methods

Gradient methods are based on the assumption that for small irrotational motions, the intensity reflected from a small patch on a sample will remain unchanged from one image to the next. That the intensity does not change due to a translation is known as the constant brightness assumption (See [18] for a more thorough discussion on the validity of this assumption).

We follow Horn's derivation [19]. He writes the constant brightness assumption as

$$\mathrm{E}(\boldsymbol{r}, t) = \mathrm{E}(\boldsymbol{r} + \boldsymbol{\delta_r}, t + \delta_t) \tag{3.1}$$

where $\mathrm{E}(\boldsymbol{r}, t)$ is the continuous brightness function and the displacement vector is $\boldsymbol{\delta_r} = (\delta_x, \delta_y, \delta_z)$. We expand the right side using a Taylor expansion to find

$$\mathrm{E}(\boldsymbol{r} + \delta_r, t + \delta_t) = \mathrm{E}(\boldsymbol{r}, t) + \nabla \mathrm{E} \cdot \boldsymbol{\delta_r} + \frac{\partial \mathrm{E}}{\partial t}\delta_t + \ldots \tag{3.2}$$

where ∇ is the gradient over spatial coordinates and the ellipsis represents high order terms. For small displacements, we drop the high order terms. (In Section 4.2, we explore what conditions make this a good approximation). Combining (3.1) and (3.2), we find

$$\nabla \mathrm{E} \cdot \boldsymbol{\delta_r} + \frac{\partial \mathrm{E}}{\partial t}\delta_t = 0. \tag{3.3}$$

Equation (3.3) is a method of measuring the local displacement vector, $\boldsymbol{\delta_r}$, occurring in time δ_t given estimates of the partial derivatives of the continuous intensity distribution.

The CCD camera samples the continuous intensity function. We call $\mathrm{G}[i, j, k, l]$ the sampled version of the intensity function taken at time l and location (i, j, k). Equation (3.3) becomes

$$\mathrm{G}_x[i, j, k, l]\delta_x + \mathrm{G}_y[i, j, k, l]\delta_y + \mathrm{G}_z[i, j, k, l]\delta_z + \mathrm{G}_t[i, j, k, l]\delta_t = 0 \tag{3.4}$$

where G_x is the estimate of the partial derivative of the continuous intensity function with respect to x, and the other partial derivatives are defined similarly. We have not specified how the partial derivatives should be calculated. Also, we have chosen to evaluate the partial derivatives at location (i, j, k, l) where the continuous intensity function was sampled; we could alternatively choose to evaluate the partial derivatives at other locations.

If the moving body is rigid and moves without rotational, the displacement vectors should be identical everywhere. By evaluating the partial derivatives at many different locations, we create a system of over-constrained equations which can be solved using the method of least squares. One minimizes the error

$$\sum_{l,m,n} \left(\mathrm{G}_x\delta_x + \mathrm{G}_y\delta_y + \mathrm{G}_z\delta_z + \mathrm{G}_t\delta_t\right)^2. \tag{3.5}$$

We differentiate to find

$$\begin{bmatrix} \widehat{\delta}_x \\ \widehat{\delta}_y \\ \widehat{\delta}_z \end{bmatrix} = - \begin{bmatrix} \sum G_xG_x & \sum G_xG_y & \sum G_xG_z \\ \sum G_yG_x & \sum G_yG_y & \sum G_yG_z \\ \sum G_zG_x & \sum G_zG_y & \sum G_zG_z \end{bmatrix}^{-1} \begin{bmatrix} \sum G_xG_t \\ \sum G_yG_t \\ \sum G_zG_t \end{bmatrix} \tag{3.6}$$

where the sums are over i, j and k and $(\widehat{\delta}_x, \widehat{\delta}_y, \widehat{\delta}_z)$ is the motion estimate.

The method of least squares provides a maximum likelihood estimate of a parameter in the presence of additive, Gaussian noise [4]. Our dominant noise sources, discussed in Chapter 2, are neither Gaussian nor additive. We therefore must regard the use of linear least squares as heuristic. However, gradient-based methods using least square techniques have been shown to be effective. Davis and Freeman shows achieves 0.01 pixel accuracy using these methods [7].

3.1.1 Higher Order Taylor Approximations

Gradient methods can be extended to higher order by taking the total derivative out to quadratic terms or further. We consider the second order expansion,

$$0 = \nabla \mathrm{E} \cdot \boldsymbol{\delta}_{\boldsymbol{r}} + \mathrm{E}_t\delta_t + \frac{1}{2}\boldsymbol{\delta}^T\nabla(\nabla\mathrm{E})\boldsymbol{\delta} + \frac{1}{2}\mathrm{E}_{tt}\delta_t^2 + \delta_t\nabla\mathrm{E}\cdot\boldsymbol{\delta} + \ldots \tag{3.7}$$

where the ellipsis represents higher order terms.

We can estimate the first and second derivatives at many points to create a set of non-linear over-constrained equations. Once again, we can use the method of least squares to solve these equations, which will result in three coupled nonlinear equations of three variables.

Finding solutions to nonlinear equations is generally difficult. However the results from the previous section can be used as a starting point for searching for the solution to this nonlinear equation. Methods of steepest descent should give acceptable results.

Higher order derivative estimates can be included in gradient-based methods by solving for the first-order estimate and then progressively solving for the higher order solutions iteratively.

3.2 Block Matching

Block Matching is based on the assumption that two images of a moving object are related by a simple shift,

$$\mathrm{G}(\boldsymbol{r}, 0) = \mathrm{G}(\boldsymbol{r} + \boldsymbol{\delta}_{\boldsymbol{r}}, 1). \tag{3.8}$$

where the 0 and 1 label the different images. The displacement between two images can then be estimated using a least square minimization,

$$(\widehat{\delta}_x, \widehat{\delta}_y, \widehat{\delta}_z) = \underset{\delta_x, \delta_y, \delta_z}{\arg\min} \sum_{i,j,k} \left(\mathrm{G}[i+\delta_x, j+\delta_y, k+\delta_z, 1] - \mathrm{G}[i,j,k,0]\right)^2. \tag{3.9}$$

To find subpixel shifts, one uses an interpolator to determine the values of the intensity function at locations that were not sampled. In principle, one could then search all possible sets of trial shifts to minimize the least square sum. However, searching is computationally intensive.

Instead, note that Equations (3.8) and (3.1) are almost identical. If one were to calculate the spatial derivatives in the gradient-based method using the interpolator of the block matching technique, the two algorithms would differ from each other in only one respect. The gradient-based method includes a time derivative that is not present in the block matching techniques.

Davis and colleagues argue that any block matching problem can be cast as an gradient-based problem [9]. By calculating the spatial and temporal derivatives of the gradient-based approach carefully, the least square minimizations in each case can be made to be identical. Similarly, Davis and colleagues show that all gradient-based approaches can be cast as block matching problems.

Since block matching and gradient method approaches are equivalent, we can choose which class of algorithms to use based on criteria such as the number of calculations required. With one exception, we choose to pursue gradient-based algorithms to avoid computationally intensive searches. The exception is Fourier Transform based block-matching methods.

3.3 Fourier Transform Based Methods

To derive the Fourier Transform motion estimator, start with the block matching Equation (3.9), and expand the square:

$$\underset{\delta_x, \delta_y, \delta_z}{\arg\min} \sum_{i,j,k} \mathrm{G}[i+\delta_x, j+\delta_y, k+\delta_z, 1]^2 - 2\mathrm{G}[i+\delta_x, j+\delta_y, k+\delta_z, 1]\mathrm{G}[i,j,k,0] + \mathrm{G}[i,j,k,0]^2 \tag{3.10}$$

Only the first and second terms depend on the translation so that we can drop the third term. Translations result in phase changes in the frequency domain. Using Parseval's theorem [34] the motion estimate becomes

$$\widehat{\boldsymbol{\delta}}_r = \underset{\boldsymbol{\delta}_r}{\arg\min} \int_{\boldsymbol{\Omega}} \left[\left|\widehat{G}^*[\boldsymbol{\Omega}, 1]\right|^2 - 2\widehat{G}[\boldsymbol{\Omega}, 0]\widehat{G}^*[\boldsymbol{\Omega}, 1] e^{-j2\pi\boldsymbol{\Omega}\cdot\boldsymbol{\delta}_r} \right] d\boldsymbol{\Omega} \tag{3.11}$$

where $\boldsymbol{\Omega}$ is the spatial frequency vector and $\widehat{G}[\boldsymbol{\Omega}, 0]$ is the Fourier transform of the windowed image $\mathrm{G}[i,j,k,0]$. Only one of the two remaining terms depends on the translation, so that we can drop the other term. The final result is the well known correlator based on the cross power spectrum, [22]:

$$\widehat{\boldsymbol{\delta}}_r = \arg\max_{\boldsymbol{\delta}_r} \int_{\boldsymbol{\Omega}} \widehat{G}[\boldsymbol{\Omega},0]\widehat{G}^*[\boldsymbol{\Omega},1]e^{-j2\pi\boldsymbol{\Omega}\cdot\boldsymbol{\delta}_r}d\boldsymbol{\Omega}. \tag{3.12}$$

We estimate the integral as a sum, evaluating the argument only at the frequencies that result from a discrete Fourier transform,

$$\widehat{\boldsymbol{\delta}}_r = \arg\max_{\boldsymbol{\delta}_r} \sum_{\boldsymbol{\Omega}} \widehat{G}[\boldsymbol{\Omega},0]\widehat{G}^*[\boldsymbol{\Omega},1]e^{-j2\pi\boldsymbol{\Omega}\cdot\boldsymbol{\delta}_r}. \tag{3.13}$$

The result suggests finding the phase-plane that best correlates with the cross power spectral density of the two images. Thus, the effect of the motion was to induce a phase-change between the spectra of the two images.

Rather than testing various translations, we fit a plane to the phase data using a least squares approach,

$$\widehat{\boldsymbol{\delta}}_r = \arg\min_{\boldsymbol{\delta}_r} \sum_{\boldsymbol{\Omega}} \left(\tan^{-1} \frac{\mathrm{Re}(\widehat{G}[\boldsymbol{\Omega},0]\widehat{G}^*[\boldsymbol{\Omega},1])}{\mathrm{Im}(\widehat{G}[\boldsymbol{\Omega},0]\widehat{G}^*[\boldsymbol{\Omega},1])} + 2\pi\boldsymbol{\Omega}\cdot\boldsymbol{\delta}_r \right). \tag{3.14}$$

Note that if the sum of the motions in the three directions is bigger than a pixel, one must "unwrap" the phase, which is one reason why many people have shunned this approach [14, 26].

We discuss this method in more detail in Chapter 6.

3.3.1 Other Fourier Transform Based Methods

There are several other methods that use the phase of a Fourier transform to estimate motion. Kuglin and Hines first developed the Phase Correlation Algorithm [25]. They examine the phase correlation function which is the inverse Fourier transform of the phase difference between two images. It can be shown that this function has a peak at the displacement between the two images. For integer shifts, this method works very well. For fractional displacement shifts, one can search for the peak in the correlation function by zero padding the Fourier transform before taking the inverse. The resolution of this method is therefore fundamentally limited by the amount of padding one is willing to do. For example, to obtain $\frac{1}{8}$ pixel resolution, one must pad by the data by a factor of 8 in all dimensions.

Others have used the phase correlation method to obtain integer pixel shifts and then used block matching based searches to determine the fractional-pixel motion [15]. Using a similar method, Pearson et al. claim they achieved 0.08 pixel accuracy [36].

Douglas developed a slightly different algorithm [11]. He starts with Equation (3.9) and uses

Parseval's theorem to write an equivalent minimization in the frequency domain:

$$\arg\min_{\boldsymbol{\delta}_r} \sum_{\boldsymbol{\Omega}} \left|\widehat{G}[\boldsymbol{\Omega}, 1] - \widehat{G}[\boldsymbol{\Omega}, 0] e^{-j2\pi\boldsymbol{\Omega}\cdot\boldsymbol{\delta}_r}\right|^2. \tag{3.15}$$

He uses a Gauss-Newton search method to find the resulting translation. By expanding Equation (3.15), one can show that the equation is identical to (3.13). However, the technique can fail to find a global minimum.

Koc and Liu derived methods to estimate subpixel motions based on the discrete cosine and sine transforms [23, 24]. They pursue this technique because the Fourier transform is incompatible with discrete cosine transform based video coding standards. They make no claims that their algorithms perform better than Fourier transform based methods. In fact, their derivation of a "pseudo-phase" simply relates cosine and sine transforms to the Fourier transform. Thus, the method is effectively a Fourier transform based method with a different name.

3.3.2 Other Basis Functions

One way to consider the Fourier transform is simply as a conversion of the data to a more useful set of basis functions. The complex exponential functions are useful since a translation simply modulates their phase.

It seems that there might be other useful basis functions. Some groups have used the Hermite Polynomials as a basis set [29]. However, their use for them was a method of calculating derivatives for interpolation.

A useful set of basis functions should contain direct information about the translation. The simplest basis set would be the set of functions which are multiplied by a factor related to the translation. These functions would be eigenfunctions of the translation operator. One can show that complex exponentials are the unique eigenfunctions of the translation operator [38].

3.4 Spatio-temporal Filters

To derive the method using of spatio-temporal filters, we start by assuming constant velocity,

$$\mathrm{E}(\boldsymbol{r}, t) = f(\boldsymbol{r} + \boldsymbol{v}t).$$

where f is the brightness function. Taking a Fourier transform in space, we find

$$\widehat{E}(\boldsymbol{\Omega}, t) = \widehat{F}(\boldsymbol{\Omega}) e^{-j2\pi\boldsymbol{\Omega}\cdot\boldsymbol{v}t}.$$

Taking a Fourier transform in time,

$$\widehat{E}(\mathbf{\Omega}, \omega_t) = \widehat{F}(\mathbf{\Omega})\delta(\omega_t + \mathbf{\Omega} \cdot \boldsymbol{v}t)$$

where δ is the Dirac delta function. For constant velocity, all energy is concentrated along a hyper-plane in the frequency domain. One might use the magnitude response to a series of bandpass filters (such as the Fourier transform) to attempt to identify the hyper-plane and thus determine the velocity. However, if motion varies in space and time, one desires a filter that is localized in regions of space and time where the velocity can be considered constant, and has a narrow pass-band in the frequency domain. Spatio-temporal filters attempt to do just that [17, 30].

The uncertainty relationship becomes important in making these filters. The uncertainty relation states that for any function, there is a limit to the product of its localization in space and its localization in the frequency domain [5]. Gabor developed a set of filters that achieve the minimum uncertainty product. Therefore, Gabor filters are often used.

For rigid body translational motion, we believe spatio-temporal filters will function adequately. However, because of the rigid body constraint, there is no need for filters that are localized in space. Therefore, we do not pursue methods using Gabor filters. Note that for rigid body rotations, Gabor methods may be useful.

The work of Fleet and Jepson using Gabor filters is particularly noteworthy because their algorithm achieves subpixel motion accuracy and consistently performs well in comparisons to other algorithms [3, 12]. To discern motion, they examine the phase of the response of the images in time to a Gabor filter. They claim that analyzing the phase yields a much more robust and a more accurate result than analyzing the magnitude of the response. Their methods are very similar to Fourier transform based methods which also detect motion using the phase response of a set of filters. Their method, however, has one major drawback: many parameters in the technique need to be experimentally determined based on the images and the motion.

3.5 Re-Sampling Methods

There are various block matching approaches based on interpolation and re-sampling of an image [36]. These techniques use interpolation to determine the value of the continuous intensity distribution between actually sampled points. These interpolated values are then treated as if they were measured. The process is known as up-sampling or re-sampling.

By up-sampling by a factor of 8, one can use very simple block matching techniques to obtain 0.125 pixel precision. These techniques effectively trade computation for memory usage. The resolution of re-sampling methods are limited by available memory and interpolation methods.

The resolution of these techniques is typically limited by memory, not by noise considerations.

Because the "new data" are functions of the initial data, the variance of the new data points is a function of the variance of the initial data. If noise were an issue, one would need to treat the interpolated data points differently than the initial data.

We note that no gain is made when re-sampling an image since the same amount of information exists before and after the re-sampling (This is the data processing inequality from information theory [6]).

3.5.1 Mutual Information Methods

Alignment can be approached as a maximization of the mutual information of two images [39]. Mutual information has many advantages over simple least square correlation techniques. Mutual information methods are insensitive to lighting gradations. These methods work for non-linear transformations. Information based techniques have reduced sensitivity to outliers in the data sets which is important when noise distributions are not exponentially distributed. Also the methods work well with multiply-peaked probability distributions, where correlation techniques fail. However, rigid body motion estimation has none of these extreme problems which mutual information based methods are useful for solving. Since mutual information methods are computationally very intensive, we do not explore them.

3.6 Summary

We examined many of the techniques in the literature used to estimate motion with the goal of finding classes of estimators that could detect the smallest possible subpixel translations of rigid bodies from a sequence of video images. We found many of the techniques to be either inappropriate for estimating rigid body translations or inheritly resolution limited. However, we noted that block matching approaches, gradient-based methods and some Fourier Transform based methods may not be inheritly resolution limited and therefore have the potential to estimate very small motions.

We saw that gradient-based methods and block matching approaches could be cast into one another. The choice of which approach to use should therefore be based on criteria such as computational considerations. We prefer gradient-based methods in order to avoid potentially expensive searches.

We choose to examine two techniques further: Fourier transform based methods and gradient-based methods.

Chapter 4

Limitations of Gradient-Based Algorithms

4.1 Introduction

We examine design limiting factors of gradient-based methods. We start by considering limitations in gradient-based methods due to using only first-order gradients in the estimate of motion. By ignoring higher order derivatives, the accuracy of gradient-based methods is limited.

If a derivative is calculated at a point (x, t) and another derivative is calculated at the same point (x, t), we call the derivatives co-located. Some researchers have published results using non-co-located derivative estimators. We examine several examples of the errors that can arise from using exact, non-co-located derivatives.

We explore the effects of errors in realizable co-located derivative estimates. We examine images of single sinusoids as test images because they are mathematically tractable. As an concrete example algorithm, we consider the first-difference estimator developed by Horn [19] and extended by Davis [10].

4.2 Accuracy Limitations of First and Second-Order Gradient-Based Methods

Typically, gradient methods use first-order derivatives to estimate motion. Ignoring higher order derivatives limits the accuracy of motion estimates. In this section, we explore the accuracy limitations in first-order and second-order gradient-based algorithms that use co-located derivative estimates. To do so, we assume that derivatives can be calculated exactly. In practice, this is not achievable. Therefore, this analysis represents an upper bound on the accuracy of an achievable gradient-based estimate.

4.2.1 Co-located, Exact, First-Order Gradients

The first-order gradient-based constraint equation was derived in section 3.1 and is given by setting the total derivative of the continuous intensity function to 0. If E is the continuous intensity function, the constraint equation is given by

$$\nabla \mathrm{E} \cdot \widehat{\boldsymbol{\delta}}_r + \frac{\partial \mathrm{E}}{\partial t}\delta_t = 0 \tag{4.1}$$

where ∇ is the spatial gradient operator, $\widehat{\boldsymbol{\delta}}_r$ is the motion estimate and δ_t is the time between two images. Consider a continuous intensity function E given by

$$\mathrm{E}(\boldsymbol{r}, t) = f(\boldsymbol{r} - \boldsymbol{d}(t)) \tag{4.2}$$

where f is a function, $\boldsymbol{r}$ is the spatial coordinate vector and $\boldsymbol{d}(t)$ is the vector displacement as a function of time. By substituting into (4.1) and calculating co-located spatial and temporal gradients, we find

$$\nabla f(\boldsymbol{r} - \boldsymbol{d}(t)) \cdot \widehat{\boldsymbol{\delta}}_r - \nabla f(\boldsymbol{r} - \boldsymbol{d}(t)) \cdot \boldsymbol{d}'(t) = \nabla f(\boldsymbol{r} - \boldsymbol{d}(t)) \cdot \left[\widehat{\boldsymbol{\delta}}_r - \boldsymbol{d}'(t)\right] = 0 \tag{4.3}$$

where $\boldsymbol{d}'(t)$ is the time derivative of the displacement function. If the gradient of f is zero, it is not possible to estimate the motion. If the gradient of f is not zero, we conclude

$$\widehat{\boldsymbol{\delta}}_r = \boldsymbol{d}'(t)\delta_t. \tag{4.4}$$

The displacement estimate is simply the first-order Taylor expansion of the displacement function, which we recognize as instantaneous velocity. Thus, first-order gradient-based methods should work well when motion has been sampled frequently enough in time so that assuming constant velocity between images is a good approximation. If the two images are taken at times 0 and 1 then the bias is given by

$$\mathrm{Bias} = \boldsymbol{d}'(t) - [\boldsymbol{d}(1) - \boldsymbol{d}(0)]. \tag{4.5}$$

The bias is the difference between the average velocity and the instantaneous velocity.

4.2.2 Co-located, Second-order Gradients

For second-order methods, we show in Appendix A that

$$\widehat{\delta}_r = \boldsymbol{d}'(t)\delta_t + \frac{1}{2}\boldsymbol{d}''(t)\delta_t^2 + O(\delta_t^4) \tag{4.6}$$

where $\boldsymbol{d}''(t)$ is the second time derivative of the displacement function.

The displacement estimate is not exactly the first two terms in the Taylor expansion of the displacement function, as one might expect. However, the error consists of terms which we assume are small in the derivation. Thus, the displacement estimate is close to what one would expect. We conclude that second-order methods have a small bias when the acceleration of an object is close to constant between images.

4.3 Effects of Non-co-located Derivative Estimates

Some researchers have used non-co-located derivatives with gradient-based methods. In this section, we explore some of the effects of calculating non-co-located spatial and temporal derivatives. For simplicity, we consider a one-dimensional image of a single spatial frequency. Once again, we assume derivatives can be calculated exactly.

Let the images have N pixels consisting of

$$\mathrm{G}[i,k] = E_0 + E_1 \cos\Omega(i+\phi-d(k)) \tag{4.7}$$

where Ω is the spatial frequency of the sinusoids, ϕ is an offset and (i,k) index the sampled spatial and temporal coordinates. If the derivatives are calculated exactly, they will be

$$\mathrm{G}_x[i,k] = -\Omega E_1 \sin\Omega(i+\phi-d(k)) \tag{4.8}$$

$$\mathrm{G}_t[i,k] = \Omega E_1 \sin\Omega(i+\phi-d(k))d'(k) \tag{4.9}$$

In one dimension, the gradient-based estimator is given by

$$\widehat{\delta}_x = \arg\min_{\delta_x} \sum_i (G_x\widehat{\delta}_x - G_t)^2, \tag{4.10}$$

which leads to

$$\widehat{\delta}_x = -\frac{\sum G_x G_t}{\sum G_x G_x}. \tag{4.11}$$

We seek to understand the effects of calculating non-co-located derivatives. We substitute $\mathrm{G}_x[i +$

$\alpha, k + \beta]$ and $\mathrm{G}_t[i, k]$, where α and β are separations in space and time, into the one-dimensional gradient-based estimator (4.11) and use trigonometry to obtain

$$\widehat{\delta_x} = d'(k) \frac{\sum_i \cos \Omega(\alpha + d(k) - d(k+\beta)) - \sum_i \cos \Omega(2i + 2\phi + \alpha - d(k+\beta) - d(k))}{N - \sum_i \cos 2\Omega(i + \alpha + \phi - d(k+\beta))}. \quad (4.12)$$

It is straight forward to show that the summation of $\cos(2\Omega i)$ over an image is small compared to the number of points in the image, N, except for frequencies in the highest and lowest $\frac{\pi}{N}$ of the spectrum. We approximate the rightmost numerator summation and the denominator summation as negligible compared to N so that we find

$$\widehat{\delta_x} \approx d'(k) \cos \Omega(\alpha + d(k) - d(k+\beta)). \quad (4.13)$$

Unless $\alpha + d(k) - d(k+\beta)$ is a multiple of 2π, the estimator will not give an estimate of velocity. The only way to guarantee an unbiased estimate of velocity is to have $\alpha = 0$ and $d(k) - d(k+\beta) = 0$.

We examine several values of α and β to see the problems that can result. We consider constant velocity motion in time to simplify the analysis. In Chapter 5 we will use methods to easily calculate derivatives at location at pixels and halfway between pixels. Therefore, we consider cases where α and β take on the values $\{-\frac{1}{2}, \frac{1}{2}, 0\}$. Since the cosine in (4.13) yields identical results independent of the sign of its argument, (α, β) and $(-\alpha, -\beta)$ yield the same errors.

In the top-right plot of Figure 4-1, derivatives are calculated at the same time, but different places ($\alpha = \frac{1}{2}, \beta = 0$ or $\alpha = -\frac{1}{2}, \beta = 0$). Estimates of motion for higher spatial frequencies give the wrong sign. One can show that the severity of the problem decreases as α shrinks towards 0.

In the top left plot derivatives are calculated at the same place, but different times ($\alpha = 0, \beta = \frac{1}{2}$ or $-\frac{1}{2}$). This algorithm gives small bias for small displacements. This method is notable because it is the basis of the standard block-matching algorithm described in Chapter 3.

In the bottom two plots, the derivatives are calculated at different times and different places ($\alpha = \pm\frac{1}{2}, \beta = \pm\frac{1}{2}$). The estimators sometimes determine the sign of the motion incorrectly. However, for the regions in which they estimate the sign of the motion correctly, the maximum bias for the highest frequencies are about 0.1 pixels. These errors are greater than those of existing algorithms [7].

4.3.1 Discussion

We have shown that if co-located temporal and spatial derivatives can be calculated perfectly then gradient-based methods exactly determine velocity. We have also shown that non-co-located derivatives can lead to errors even if the derivatives are calculated perfectly. We have not excluded the possibility of the existence of a small bias, non-co-located, gradient-based motion estimator. How-

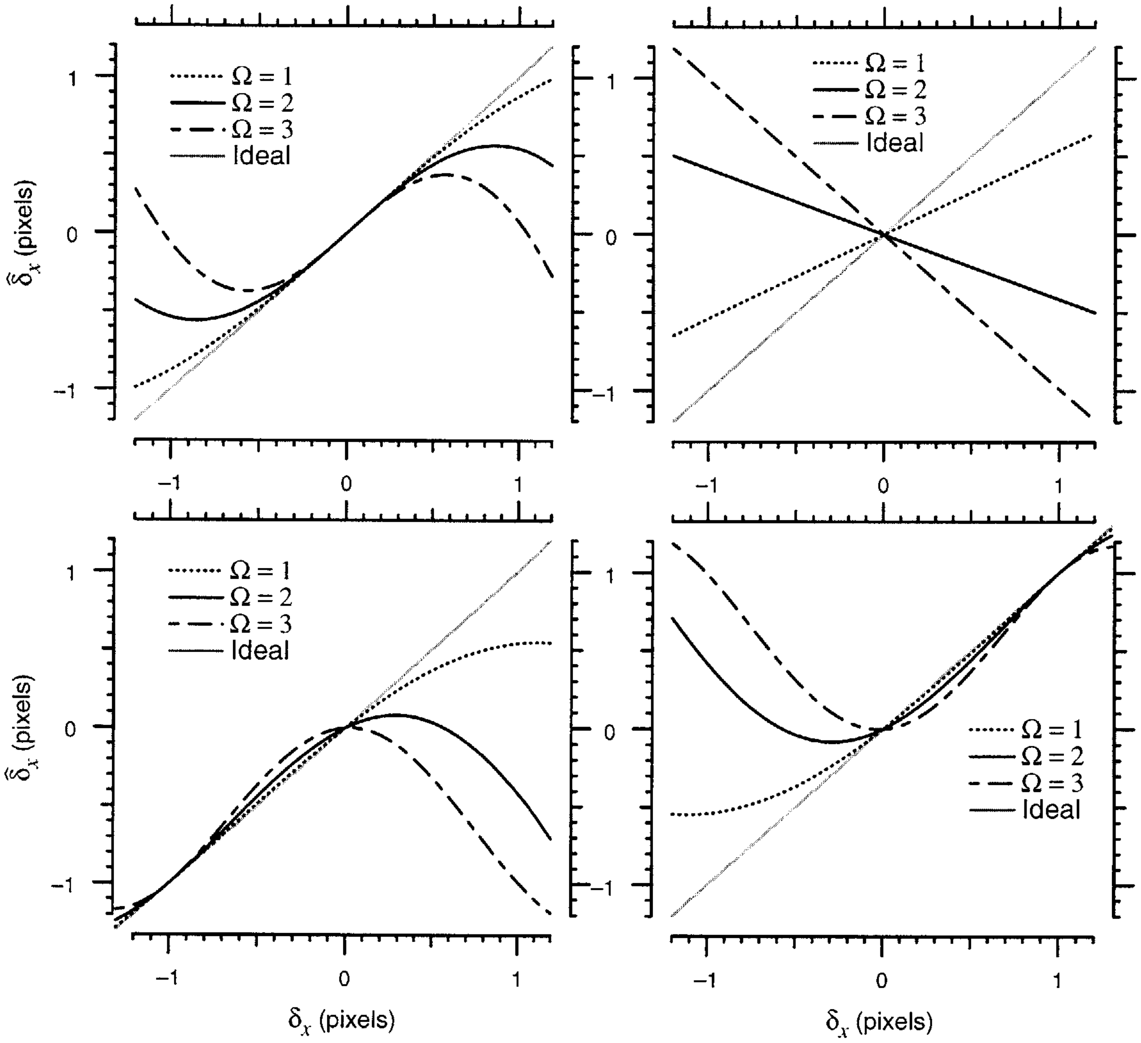

Figure 4-1: The effect of calculating exact, non-co-located spatial and temporal derivatives for linear motion in time. For the top right, the derivatives are separated by $(\alpha = \frac{1}{2}, \beta = 0)$. For the top left, the derivatives are spaced by $(0, \frac{1}{2})$. In the bottom left plot, the derivatives are spaced by $(\frac{1}{2}, \frac{1}{2})$ and for the bottom right the derivatives are spaced by $(-\frac{1}{2}, \frac{1}{2})$.

ever, we argue that any researcher attempting to develop a non-co-located gradient-based method should be careful to understand the limitations of the algorithm. We consider only co-located derivative estimates in the remainder of this chapter.

4.4 Co-located, Realizable Derivatives

In this section, we examine the effects of errors in realizable co-located derivative estimates using linear functions of the data. We examine images of single sinusoids in one dimension because these images are mathematically simple enough to obtain closed-form expressions for motion estimates.

4.4.1 Derivatives Estimates as Filters

Linear functions can be considered filters. We consider derivative estimates as a cascade of one-dimensional filters in space and time convolved with a sequence of video images. We choose one-dimensional filters rather than multidimensional filters to simplify the analysis.

In one dimension, a spatial derivative estimate, for example, consists of a spatial derivative filter and a temporal interpolation filter. The interpolation may be necessary so that derivatives are co-located. The spatial and temporal derivative estimates, G_x and G_t, can be written as

$$\mathrm{G}_x = H_x[i] * H_{it}[t] * \mathrm{G}[i,t], \text{ and} \tag{4.14}$$

$$\mathrm{G}_t = H_{ix}[i] * H_{it}[t] * \mathrm{G}[i,t] \tag{4.15}$$

where H_l is a derivative filter in the l^{th} direction and H_{il} is an interpolating filter in the l^{th} direction. The phase lag of each filter determines the location of the derivative estimate. The filters are chosen to have linear phase so that the contributions of each spatial frequency to a derivative estimate are in phase. In the frequency domain, the requirement of linear phase limits the form of each filter, so that

$$H_x(\Omega_x,\Omega_t) = D_x(\Omega_x)e^{-j\Omega_x\alpha_x}, \tag{4.16}$$

$$H_{ix}(\Omega_x,\Omega_t) = D_{ix}(\Omega_x)e^{-j\Omega_x\alpha_{ix}}, \tag{4.17}$$

$$H_t(\Omega_x,\Omega_t) = D_t(\Omega_t)e^{-j\Omega_t\alpha_t}, \text{ and} \tag{4.18}$$

$$H_{it}(\Omega_x,\Omega_t) = D_{it}(\Omega_t)e^{-j\Omega_t\alpha_{it}}, \tag{4.19}$$

where the D's are real functions and the α's are phase lags. So that the derivative estimates are co-located, the phase lag of the spatial filters in each derivative estimator must be identical. Similarly the phase lag of the temporal filters of each derivative estimator must be the same. Therefore $\alpha_x = \alpha_{ix}$ and $\alpha_t = \alpha_{it}$.

4.4.2 Constant Velocity Motion in One Dimension

We consider the sequence of one-dimensional images of a sinusoid moving with constant velocity

$$G[i,t] = E_1 \cos[\Omega(i + \phi - \delta_x t)] \tag{4.20}$$

where t indexes the images in time. The only spatial frequency in these images is Ω. Similarly, the only temporal frequency in these images in $-\Omega\delta_x$. We find that the derivative estimates due to the filters are given by

$$G_x[i] = D_x(\Omega) D_{it}(-\Omega\delta_x) E_1 \cos\Omega(i + \phi - \alpha_x - \delta_x(t - \alpha_t)), \text{ and} \tag{4.21}$$

$$G_t[i] = D_{ix}(\Omega) D_t(-\Omega\delta_x) E_1 \cos\Omega(i + \phi - \alpha_x - \delta_x(t - \alpha_t)). \tag{4.22}$$

Using 4.11, we find the gradient estimate of motion to be

$$\widehat{\delta}_x = -\frac{D_{ix}(\Omega)}{D_x(\Omega)} \frac{D_t(-\Omega\delta_x)}{D_{it}(-\Omega\delta_x)}. \tag{4.23}$$

The estimate of motion is always independent of the phase of the sinusoid and the number of places at which the derivatives are evaluated. Also, the estimate of the motion depends only on the ratios of the filters along the space and time axes, not on the filters themselves. Observe that if identical derivative and interpolation filters are used in space and time, that is if $D_x = D_t$ and $D_{ix} = D_{it}$, for translations between adjacent images of 1 pixel, the estimator is unbiased.

4.4.3 Example: The First-Difference Algorithm in One Dimension

Consider two sampled images $\mathrm{G}[i,0]$ and $\mathrm{G}[i,1]$ acquired at times 0 and 1, and over the spatial range $i = 1, 2, \ldots, N$. For two images, one can assume constant velocity motion without loss of generality so that the results of the last section apply. The first-difference derivative estimators are given by

$$\mathrm{G}_t[i] = \frac{1}{2}\sum_{i'=i}^{i+1} \mathrm{G}[i',1] - \mathrm{G}[i',0], \text{ and} \tag{4.24}$$

$$\mathrm{G}_x[i] = \frac{1}{2}\sum_{k'=0}^{1} \mathrm{G}[i+1,k'] - \mathrm{G}[i,k']. \tag{4.25}$$

The method calculates the spatial derivative by taking a first-difference between two adjacent points in each of the images and then averaging the two results together. This spatial derivative calculation is equivalent to using the 2-point derivative filter $(1, -1)$ in each image and then using the interpolating filter $(\frac{1}{2}, \frac{1}{2})$ in time. Similarly, the time derivative estimate is equivalent to using the 2-point derivative filter $(1, -1)$ in time and then the 2-point interpolating filter $(\frac{1}{2}, \frac{1}{2})$ in space.

For the first-difference algorithm we find,

$$\begin{aligned}
&D_t(-\Omega\delta_x) = 2\sin\tfrac{\Omega\delta_x}{2},\\
&D_{ix}(\Omega) = \cos\tfrac{\Omega}{2},\\
&D_x(\Omega) = -2\sin\tfrac{\Omega}{2}, \text{ and}\\
&D_{it}(-\Omega\delta_x) = \cos\tfrac{\Omega\delta_x}{2},
\end{aligned}$$

The derivative of $e^{-j\Omega x}$ is $-j\Omega e^{-j\Omega x}$ so that the magnitude of the frequency response of an ideal differentiator is Ω. In Figure 4-2, the frequency response of the first-difference derivative filter is compared to the frequency response of the ideal derivative estimator. For large frequencies the derivative estimator diverges from the ideal result. An ideal interpolator shifts a complex exponential, leaving its magnitude unchanged. Therefore, an ideal interpolator has an amplitude response of one. In Figure 4-2, we show that the response of the 2-point interpolator diverges from the response of the ideal interpolator at very small frequencies.

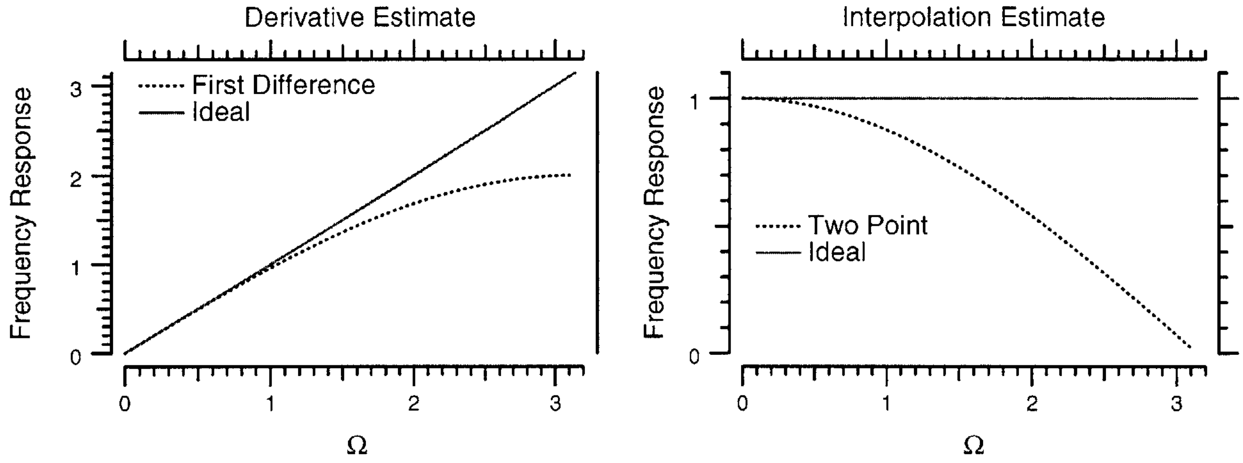

Figure 4-2: Frequency response of first-difference derivative filter and interpolation filter compared with the frequency response of the corresponding ideal filters.

Using Equation (4.23), we find the first-difference estimate of motion for images of single frequencies to be

$$\widehat{\delta}_x = \frac{\tan(\Omega\delta_x/2)}{\tan(\Omega/2)} \tag{4.26}$$

which is plotted in Figure 4-3. For low frequency images, the estimator is almost unbiased. For $\Omega = 1$, the method makes errors near hundredths of a pixel. For $\Omega = 2$, the bias of the first-difference algorithm is closer to tenths of a pixel. For $\Omega = 3$, the bias in the algorithm is close to 0.5 pixels.

Given that the 2-point interpolator diverges from the ideal interpolator at such low frequencies, one might expect that the first-difference algorithm would make much larger errors than it actually

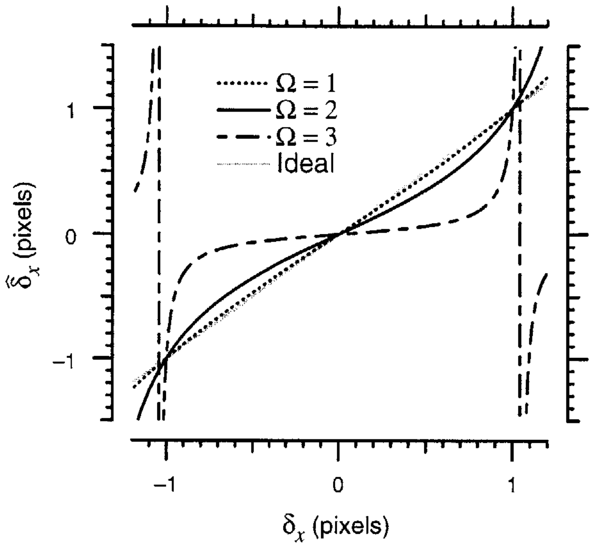

Figure 4-3: Estimates $\widehat{\delta}_x$ of the displacement δ_x between two sinusoidal images using the first difference estimator. The curves illustrate gradient-based estimates of displacement (4.11) for sinusoidal images with frequencies Ω, where $\Omega = \pi$ is the Nyquist frequency.

does. However, Equation (4.23) shows that it is not the performance of the individual filters that is important but the ratio of the filters. The ratios of the filters are given by

$$\frac{H_{ix}(\Omega)}{H_x(\Omega)} = \frac{1}{2\tan\frac{\Omega}{2}}, \text{ and} \tag{4.27}$$

$$-\frac{H_t(-\Omega\delta_x)}{H_{it}(-\Omega\delta_x)} = 2\tan\frac{\Omega\delta_x}{2}. \tag{4.28}$$

In Figure 4-4, one can see that the ratio of the first difference differentiator to the 2-point interpolator is close to ideal up to frequencies near 1. In some sense, the error in the interpolator was somewhat counteracted by the error in the differentiator.

4.4.4 Sinusoidal Motion in One Dimension

In this section, we show that the filter-based techniques we have developed apply to more general motions. We consider the case of sinusoidal motion in time. Consider a sequence of images of the form:

$$G[i,t] = \cos\Omega_x(i + \phi_x - A_x d), \text{ with}$$
$$d = \sin\Omega_t(t + \phi_t)$$

where A_x is the amplitude, ϕ_t is the phase and Ω_t is the frequency of the sinusoidal motion. Filters in space act independently of the motion in time so that we need only determine the effect of filters in time. First, we determine the temporal frequencies in the image. To do this, we expand the cosine

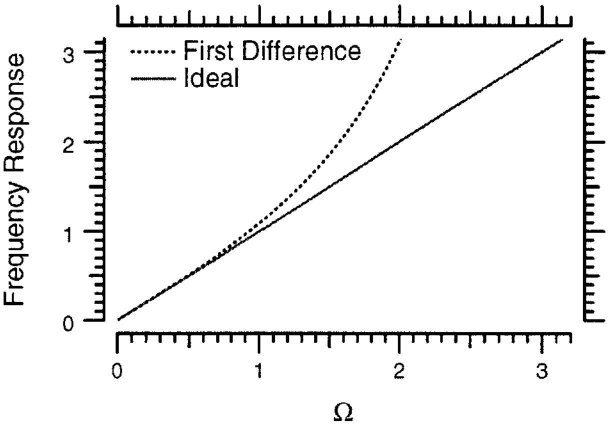

Figure 4-4: Plot of $2\tan\frac{\Omega}{2}$ which is the ratio of the frequency response of the 2-point differentiator $(1,-1)$ to the 2-point interpolator $(\frac{1}{2},\frac{1}{2})$.

in a Taylor series in the displacement function,

$$\begin{aligned}\cos\Omega_x(i+\phi_x-A_xd) = \cos[\Omega_x(i+\phi_x)] - \sin[\Omega_x(i+\phi_x)](\Omega_xA_xd)& \\ +\frac{1}{2}\cos[\Omega_x(i+\phi_x)](A_x\Omega_xd)^2+\dots&\end{aligned} \tag{4.29}$$

where the ellipsis represents higher order terms. If a filter in time H_t with response $D_t(\Omega_t)$ acts on the image, we find

$$\begin{aligned}D_t\cos(\Omega_x(i+\phi_x-A_xd)) = D_t(0)\cos(\Omega_x(i+\phi_x)) - D_t(\Omega_t)(\Omega_xA_xd)\cos(\Omega_x(i+\phi_x+\frac{\pi}{2}))& \\ +D_t(2\Omega_t)\frac{(\Omega_xA_xd)^2}{2}\cos(\Omega_x(i+\phi_x))+\dots&\end{aligned} \tag{4.30}$$

where we have assumed no phase lag for simplicity. Notice that the result contains the same spatial frequencies in the original image, though some of the frequencies are shifted by $\frac{\pi}{2}$. Cosines of the same frequency with relative phase lag of $\frac{\pi}{2}$ are approximately orthogonal when summed over an image. Using this approximation, we find the motion estimate,(4.11), to be

$$\hat{\delta}_x = -\frac{(H_{it}(0)+H_{it}(2\Omega_t)\frac{(\Omega_xA_xd)^2}{2})(H_t(0)+H_t(2\Omega_t)\frac{(\Omega_xA_xd)^2}{2})+H_{it}(\Omega_t)H_t(\Omega_t)(\Omega_xA_xd)^2}{(H_{it}(0)+H_{it}(2\Omega_t)\frac{(\Omega_xA_xd)^2}{2})^2+H_{it}(\Omega_t)^2(\Omega_xA_xd)^2}\,\frac{H_{ix}(\Omega_x)}{H_x(\Omega_x)} \tag{4.31}$$

where we have included only terms up to the second-order in the expansion.

4.4.5 Constant Velocity Motion in Two Dimensions

In two dimensions, the gradient-based estimator is given by

$$\widehat{\delta}_x = \frac{\sum G_xG_y \sum G_yG_t - \sum G_y^2 \sum G_xG_t}{\sum G_x^2 \sum G_y^2 - \sum G_xG_y \sum G_xG_y} \tag{4.32}$$

$$\widehat{\delta}_y = \frac{-\sum G_x^2 \sum G_yG_t + \sum G_xG_y \sum G_xG_t}{\sum G_x^2 \sum G_y^2 - \sum G_xG_y \sum G_xG_y}. \tag{4.33}$$

In two dimensions, as in one dimension, we consider derivatives as cascades of filters. Derivative estimates consist of a cascade of two interpolating filters and one differentiating filter. We consider the sequence of images given by

$$G[i,j,t] = E_1 \cos\left[\Omega_x(i+\phi_x-\delta_x t) + \Omega_y(j+\phi_y-\delta_y t)\right]. \tag{4.34}$$

Our choice of test images leads to difficulties because each image in the sequence is constant along lines parallel to $\Omega_x i + \Omega_y j = 0$. Motion cannot be determined along this direction. We have chosen this sequence of images despite this difficulty because of the particularly simple form of the motion estimates. Motions can be discerned perpendicular to the lines of constant intensity. This direction is constrained to have $\Omega_x\widehat{\delta}_y = \widehat{\delta}_x\Omega_y$. Solving Equation (4.32) using this constraint, we find the component vectors of the motion to be

$$\widehat{\delta}_x = \frac{-H_t(-\Omega_t)}{H_{it}(-\Omega_t)} \; \frac{\Omega_x}{\frac{H_x(\Omega_x)}{H_{ix}(\Omega_x)}\Omega_x + \frac{H_y(\Omega_y)}{H_{iy}(\Omega_y)}\Omega_y}, \text{ and} \tag{4.35}$$

$$\widehat{\delta}_y = \frac{-H_t(-\Omega_t)}{H_{it}(-\Omega_t)} \; \frac{\Omega_y}{\frac{H_x(\Omega_x)}{H_{ix}(\Omega_x)}\Omega_x + \frac{H_y(\Omega_y)}{H_{iy}(\Omega_y)}\Omega_y} \tag{4.36}$$

where $\Omega_t = \Omega_x\delta_x + \Omega_y\delta_y$. Once again, the motion estimates depend only on the ratio of filters in each direction.

4.4.6 The First-Difference Algorithm in Two Dimensions

We explore the first-difference algorithm is two dimensions to analyze crosstalk. We consider the images

$$G[i,j,0] = \cos\Omega_x(i+\phi_x)\cos\Omega_y(j+\phi_y) \tag{4.37}$$

$$G[i,j,1] = \cos\Omega_x(i+\phi_x-\delta_x)\cos\Omega_y(j+\phi_y-\delta_y) \tag{4.38}$$

In two dimensions, the first-difference derivatives consist of a 2-point derivative and two 2-point interpolators; the formulas are given in [7]. After a a lengthy calculation, we find[1].

$$\widehat{\delta}_x = \frac{1}{\tan\frac{\Omega_x}{2}} \frac{\sin\Omega_x\delta_x}{\cos\Omega_y d_y + \cos\Omega_x\delta_x} \tag{4.39}$$

$$\widehat{\delta}_y = \frac{1}{\tan\frac{\Omega_y}{2}} \frac{\sin\Omega_y d_y}{\cos\Omega_y d_y + \cos\Omega_x\delta_x} \tag{4.40}$$

The motion estimates are shown in Figure 4-5. The error in motion estimates in the x-direction increases as $\Omega_y\delta_y$ increases for constant Ω_x. The sign of the motion estimate is not always correct. For some combinations of frequencies and displacements, the estimates of motion are infinite.

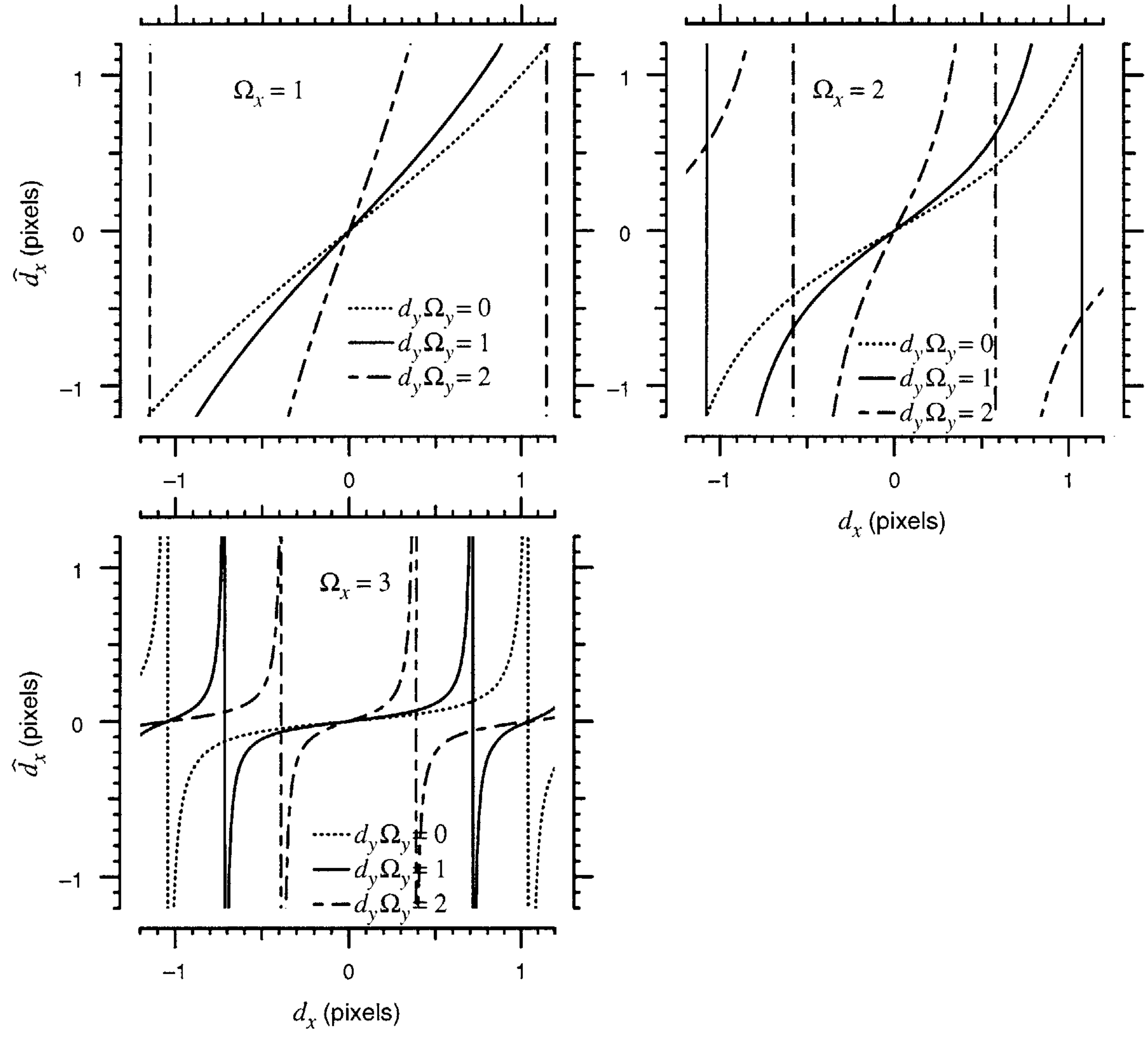

Figure 4-5: Crosstalk between motion in two directions for an image consisting of a spatial frequency Ω_x in the x direction and Ω_y in the y-direction.

[1] We simulated the results and found that the formulas agreed with the simulated results in all six decimal places examined.

4.4.7 Discussion

In all cases examined, we found that the spatial filters used in the estimator in each direction appeared as ratios of interpolators to differentiators. That this happened is very general, independent of the spectral content of the image. To understand this, consider prefiltering an image along the x-direction. Rather than prefiltering the image, one could equivalently convolve all the derivative and interpolation filters in the x-direction which the prefilter. In the frequency domain, the ratio of the response of derivative and interpolating filters along that direction would remain unchanged. However, the frequency response of each filter may look very little like the ideal derivative and interpolating filters. We conclude that when designing gradient-based estimators, one need not worry about the exact shape of derivative and interpolation filter in space, only their ratio.

If derivative and interpolation filters are not ideal, they can be considered the convolution of the ideal filters and a prefilter of the image. For the first-difference algorithm, the derivative and interpolation filters both decrease from an ideal as shown in Figure 4-2. Thus, the first-difference algorithm effectively includes a low pass filter.

Examining Figure 4-3, the estimate of motion for $\Omega = 3$ becomes infinite for displacements slightly larger than one pixel. This effect illustrates a problem of the gradient-based methods. For large motions, gradient-based methods cease to give accurate motion estimates. We pointed out that for images of a single spatial frequency in one dimension, the only temporal frequency in the image is $\Omega\delta_x$. Once this temporal frequency becomes larger than π, temporal frequencies are aliased. For $\Omega = 3$, aliasing in the time domain occurs for displacements slightly larger than one pixel.

For sinusoidal motion, terms of d^2 and d^3 lead to higher harmonics of the temporal frequency Ω_t. For high order terms, the higher harmonics will become greater than π. Those harmonics will be aliased into the other temporal frequencies. For large motion, we expect the energy in the higher harmonics to be important and therefore the effects from aliasing will become important. Examining the expansion in Section 4.4.4, "large" means $A_x\Omega_x$ approaching 1.

One might presume that the poor estimates of motion of the first difference algorithm in two dimensions is due to poor derivative and poor interpolation estimates. However, if that were the case, similar effects would have been seen in one dimension. We pointed out in Section 4.4.5 that the temporal frequencies in the image are given by $\Omega_t = \Omega_x\delta_x + \Omega_y\delta_y$. Motion in both directions lead to higher temporal harmonics than motion in only one direction. We attribute the poor estimates of motion in Figure 4-5 to aliasing temporal frequencies greater than π.

4.5 Relative Errors in Filters

It is not possible to calculate derivatives exactly. An important question to consider is the relative importance of temporal and spatial filter errors. In data collected using video microscopy, one typically has many more points in an image for the support of spatial filters than images in time for the support of temporal filters. The question arises if it is worth while to calculate spatial derivatives to much better accuracy that temporal derivatives.

For simplicity, we consider the case of constant velocity motion in one dimension for images of a single sinusoid. It is well known that for uncorrelated errors, the square of the fractional error in a ratio is given roughly by the sum squares of the fractional error of the elements in the ratio[2][4]. Using Equation (4.23), we immediately conclude

$$\frac{\epsilon_{\widehat{\delta}_x}}{\widehat{\delta}_x} \approx \left((\frac{\epsilon_{D_t/D_{it}}}{D_t/D_{it}})^2 + (\frac{\epsilon_{D_{ix}/D_x}}{D_{ix}/D_x})^2 \right)^{\frac{1}{2}} \tag{4.41}$$

where the epsilons represent errors in the corresponding ratios.

Thus if the fractional error in any one of the ratios is larger that the rest, its square will dominate the sum. Similar results can be derived in higher dimensions. We conclude that there is little advantage to calculating filters in one dimension to much higher accuracy than filters in any other dimension.

4.6 Conclusions

We showed that estimating co-located derivatives in first-order gradient-based methods leads to estimates of velocity. We showed that for spatial filters used in gradient-based methods, it is the ratio of the response of the derivative and interpolation filters that is key, not the response of the individual filters themselves.

We showed that for large motions, gradient-based methods can be expected to produce poor estimates of motion. We showed that when motions become large enough, temporal frequencies greater than π are produced and aliased to other frequencies. We showed that these error can cause the gradient-based estimates to become arbitrarily large.

We also concluded that calculating spatial filters to much higher accuracy than temporal filters is only marginally productive. We conclude that to calculate gradient-based methods to high accuracy, we need to calculated spatial and temporal filters to high accuracy. This conclusion leads us toward using sequences of images in the support of temporal filters. We investigate multi-image based gradient methods in the next chapter.

[2] Actually, this result is typically derived assuming the elements in the ratio are noisy. However, we can equally well assume that the elements in the ratio are equal to their mean plus some small error.

Chapter 5

Development and Simulations of Multi-Image Gradient-Based Algorithms with Applications for Sinusoidal Motion Estimation

5.1 Abstract

We demonstrate a design method for motion detection algorithms that use data in many images to estimate motion between two images. We use the technique to develop a class of gradient-based algorithms to measure the amplitude and phase of an object undergoing sinusoidal motion. We simulate the performance of the algorithm in the absence of noise to show that the systematic bias in the class of amplitude estimators is less than 0.001 pixels for motions smaller than 1.2 pixels in amplitude. For our applications in which noise is of the order of −50 dB, amplitude estimates with 0.001 precision and accuracy and phase estimates with 0.001 radian precision and accuracy are achievable.

5.2 Introduction

Davis and Freeman [7] used the first difference algorithm created by Horn [19, 18] to perform sub-pixel image registration. They found the first difference algorithm bias limited and created a linear bias compensation (LBC) algorithm to improve the performance of the registration method. Still, they found the bias inherent to LBC to be roughly one order of magnitude higher than the standard deviations due to noise. In the presence of blur due to non-zero image acquisition times, the errors inherent to LBC increase [8].

We create a technique for designing gradient-based algorithm whose estimates, for small enough motions, are not limited by errors inherent to the algorithm. To improve the accuracy of gradient-based algorithms over the first difference algorithm, it is not sufficient to improve spatial derivative estimates alone; it is necessary to improve both spatial and temporal derivatives estimates (see Chapter 4 or Chapter 6 of [10]). To improve derivative calculations, we use the data in a sequence of images rather than two images. Additionally, by using data in a sequence of images, we are able to compensate for effects due to finite image acquisition times.

We are motivated by our work using optical microscopy to measure the driven motions of biomechanical and microelectromechanical systems (MEMS). When exciting many of these systems with sinusoids, we expect sinusoidal responses of the same frequency. Therefore, we use our technique to develop a least square estimator of amplitude and phase of temporal sinusoidal motion.

5.3 The Design of Multi-Image Gradient-Based Algorithms

Let $G[i,j,t]$ represent a time-series of sampled images. We showed in Chapter 3 that the gradient-based estimate of motion between two adjacent images in the sequence is given by

$$\begin{bmatrix} \widehat{\delta}_x \\ \widehat{\delta}_y \\ \widehat{v}_x \\ \\ \widehat{v}_y \end{bmatrix} = - \begin{bmatrix} \sum\sum \mathrm{G}_x\mathrm{G}_x & \sum\sum \mathrm{G}_x\mathrm{G}_y \\ \sum\sum \mathrm{G}_x\mathrm{G}_y & \sum\sum \mathrm{G}_y\mathrm{G}_y \end{bmatrix}^{-1} \begin{bmatrix} \sum\sum \mathrm{G}_x\mathrm{G}_t \\ \sum\sum \mathrm{G}_y\mathrm{G}_t \end{bmatrix} \tag{5.1}$$

where the sums are over i and j, $(\mathrm{G}_x, \mathrm{G}_y, \mathrm{G}_z, \mathrm{G}_t)$ are the estimates of the spatial and temporal derivatives of the continuous intensity function, and $(\widehat{\delta}_x, \widehat{\delta}_y)$ is the motion estimate. We then showed in Chapter 4 that if the co-located derivatives are calculated exactly, the motion estimates measure instantaneous velocity, not displacement.

Multi-image gradient-based methods use data in a sequence of images to estimate the gradients in Equation (5.1). The challenge of finding a gradient-based estimate of velocity is finding estimates of co-localized derivatives and choosing the points at which the derivatives will be calculated.

5.3.1 Derivative Estimation

We choose to calculate derivatives using linear functions of the data so that unbiased noise in the image acquisition process does not create a bias in derivative estimates. Since linear functions can be considered filters, the process of calculating derivatives is equivalent to finding a derivative-estimating filter.

We consider a derivative estimate as a convolution of the data with a cascade of one-dimensional filters in each direction. While there are potential advantages to creating multidimensional filters rather than sequences of one dimensional filters, for simplicity of design we choose to create one dimensional filters. In our formulation, the most general derivative calculation in the x-direction, for example, consists of a derivative filter in the x-direction, an interpolation filter in the y-direction, and an interpolation in time. The interpolations may be necessary so that all derivatives are co-located. The derivative estimates can be written as

$$\begin{aligned} \mathrm{G}_x[i+\alpha_i, j+\alpha_j, t+\alpha_t] &= H_x[i] * H_{iy}[j] * H_{it}[t] * \mathrm{G}[i,j,t] \\ \mathrm{G}_y[i+\alpha_i, j+\alpha_j, t+\alpha_t] &= H_y[i] * H_{ix}[j] * H_{it}[t] * \mathrm{G}[i,j,t] \\ \mathrm{G}_t[i+\alpha_i, j+\alpha_j, t+\alpha_t] &= H_t[i] * H_{iy}[j] * H_{ix}[t] * \mathrm{G}[i,j,t] \end{aligned}$$

where H_l is a derivative filter in the l^{th} direction, H_{il} is an interpolating filter in the l^{th} direction, and $(\alpha_i, \alpha_j, \alpha_t)$ represents the offset relative to a pixel location where the derivatives are calculated. The delay of the filters determines the locations at which the derivatives are estimated. The filters are chosen to have linear phase so that the contributions of each spatial frequency to a derivative estimate are in phase.

We choose to estimate derivatives at one of two locations relative to a pixel (i,j) in space: at $(i+\frac{1}{2}, j+\frac{1}{2})$ and at the pixel (i,j). We calculate temporal derivatives either at the time an image was taken, or between two successive images

5.3.2 Filter Design Using The Parks McClellan Algorithm

We use the Parks-McClellan algorithm (reviewed in [34]) to create the one dimensional filters in space and time. We briefly review the algorithm here. The method requires a desired filter response $H_d(\omega)$, the desired length of the finite impulse response filter, and a weighting function $W(\omega)$ indicating the relative importance of errors as a function of frequency. The Parks McClellan Algorithm determines the filter $A(\omega)$ that minimizes the maximum of the absolute value of the weighted error $E(\omega)$, given by

$$|E(\omega)| = |W(w)[H_d(\omega) - A(\omega)]|. \tag{5.2}$$

The algorithm achieves the minimum by creating a filter whose error oscillates from positive to negative with equal weighted amplitude.

To use the algorithm, we must determine the appropriate error weighting function. We would like the error for a typical image to be as small as possible. For an image with spectrum G(ω) along one direction, the error is given by

$$\int \mathrm{G}(\omega)[H_d(\omega) - A(\omega)]d\omega = \int \mathrm{G}(\omega)\frac{E(\omega)}{W(\omega)}d\omega. \tag{5.3}$$

Since $E(\omega)$ oscillates from positive to negative, one way make the expected error small is to use the spectrum of an "average" image as the weighting function in the Parks McClellan algorithm. For our microscope images, we choose an ω^{-1} weighting.

We find there is little energy in our microscope images above spatial frequency 2. In order to obtain smaller errors for frequencies smaller than 2, we set the weighting function to be 0 above spatial frequency 2.

All filters were calculated using Matlab 5.1's "remez" function and are listed in the last section of this chapter.

5.3.3 Derivative and Interpolation Filters

The derivative of $e^{-j\omega x}$ is $-j\omega e^{-j\omega x}$ so that the magnitude of the frequency response of an ideal differentiator is ω. Derivative filters have odd symmetry so that filters which estimate derivatives at a pixel have an odd number of points in their support. Filters which estimate derivatives between pixels have an even number of points. Several example derivative filters are shown in Figure 5-1.

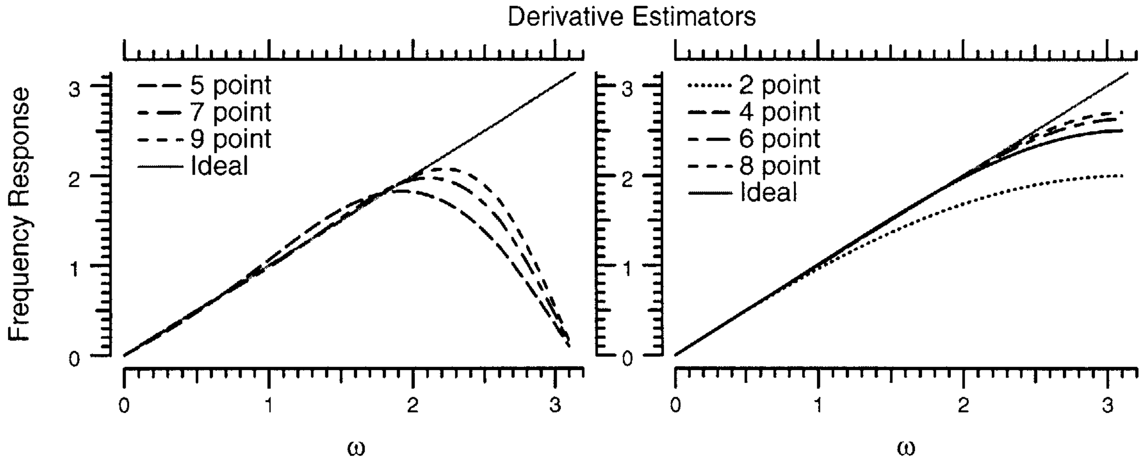

Figure 5-1: Frequency response of odd and even derivative filters created using the Parks McClellan algorithm with weighting function that varies inversely with frequency in the range 0 to 2 and is 0 in the range 2 to π. The filter with support 2 is used in the first difference algorithm.

By using odd filters to estimate all the gradients at the same point (i, j), no interpolating filters are necessary. Conversely, even derivative estimators in the x-direction, for example, estimate a derivative at $(i + \frac{1}{2}, j)$ so that an interpolator in the y-direction is necessary so that all derivatives are estimated at $(i + \frac{1}{2}, j + \frac{1}{2})$. Filters that interpolate $\frac{1}{2}$ pixel are symmetric with even support.

An ideal interpolator has a magnitude response of 1. Figure 5-2 illustrates several example filters.

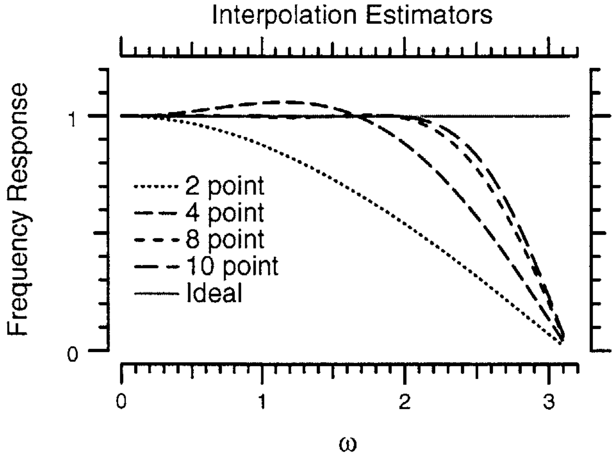

Figure 5-2: Frequency response of interpolation filters created using the Parks McClellan algorithm with weighting function as described in Figure 5-1. The filter with support 2 is used in the first difference algorithm.

In Figure 5-3, the maximum of the weighted error, Equation (5.2), is plotted versus filter support for each of the three types of filters. The errors decrease exponentially with filter support.

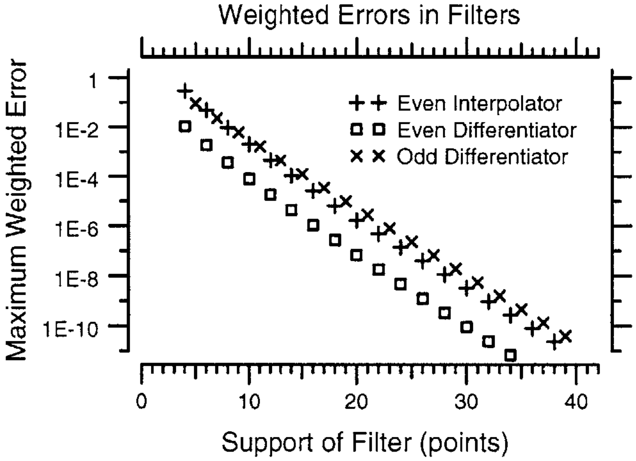

Figure 5-3: Maximum weighted error (5.2) of filters created using a Parks McClellan algorithm with weighting function as described in Figure 5-1.

5.3.4 Example: Constant Velocity Motion in One Dimension

As a simple example, we consider a gradient-based estimator in one dimension analyzing an image of a single sinusoid, translating with constant velocity. As shown in Chapter 4, the number of points in the image and the phase of the sinusoid have no effect on the results.

We consider the sequence of images $G[i, t] = \sin 2(i - vt)$. This choice should give the largest error for our estimators in the frequency range of interest. We use identical derivative filters in space and time and identical interpolation filters in space and time. Figure 5-4 shows the bias in the estimate of the motion between 2 images using the multi-image gradient-based algorithms with varying support. The error in the motion estimate drops exponentially with the number of images and number of points in space that are used. Note the very slight trend of increasing error for larger velocities.

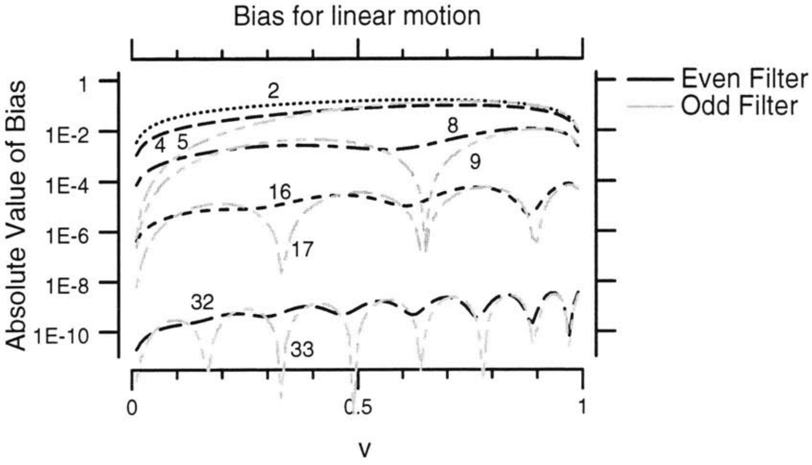

Figure 5-4: Error in the estimate of motion between two images using the data in many images. The sequence of images consists of $\sin 2(i - vt)$. The bottom two traces use 32 images and 32 points in space and 33 images and 33 points in space to calculate each derivative. The next highest traces use 16x16 and 17x17 regions in space-time. The number of images and pixels used to calculate each derivative in an estimator is shown next to its trace. The highest trace is the first difference algorithm with a support of 2x2.

5.3.5 Discussion

Figure 5-3 shows that the weighted error of even derivative filters is more than an order of magnitude smaller than the weighted error of odd derivative filters for supports of each filter that differ by one. Interpolation filter errors with even support, however, are comparable to odd derivative filter errors. The result is that, for the example shown in Figure 5-4, the bias of estimators based on even filters is comparable to the bias of estimators based on odd filters.

As shown in Figure 5-3, the error in the filters decrease exponentially with filter support. This decrease in error leads to an exponential decrease in error in motion estimation with filter support, as shown in Figure 5-4. That the motion error decreases exponentially with filter support illustrates the huge potential advantages of using multi-image based methods.

It is straightforward to show that for the example, $2v$ is the only temporal frequency in the images we examine. As v approaches 1, the temporal frequency approaches 2 and the error in the temporal filters increases due to the increased ripple allowed by the Parks McClellan algorithm. The increasing ripple explains the very slight trend of increasing error for larger velocities shown in

Figure 5-4.

If v were to continue increasing, eventually the temporal frequency would pass π. Temporal frequencies higher than π are aliased into other frequencies. Thus, the multi-image gradient methods should cease to estimate motion well for large enough motions. It is generally true that multi-image gradient-based methods give small errors only for small enough motions.

5.4 Development of a Sinusoidal Motion Estimator

We develop a multi-image gradient-based estimator of amplitude and phase for motion that is believed to be sinusoidal at a known frequency. To do this, we specify the design of the filters and determine how to combine individual estimates of velocity into an estimator of amplitude and phase. For concreteness, we assume that eight evenly-spaced images are collected per period of the motion. We allow the temporal filters to wrap around, using the first picture as if it followed the last and the last picture as if it preceded the first.

5.4.1 Derivative design

The spatial frequencies in an image are independent of the motion. Therefore, for the design of spatial derivative and interpolation filters, we use the same error weighting function that is used in Section 5.3.2, ω^{-1} between spatial frequencies 0 and 2, and 0 between 2 and π.

Although the motion is sinusoidal, the changes in brightness are not. However, the changes in brightness are periodic, consisting of a linear combination of the oscillating frequency and its higher harmonics. The temporal filters need be accurate only in a small bandwidth around those frequencies. For the case of eight images per cycle, those frequencies are $\frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi$ where π is the Nyquist frequency. We defined the region of non-zero weighting function around each frequency ω to be 0.999ω to 1.001ω, except for π for which the region is 0.999π to π. It is not clear what weighting function is appropriate. We used ω^{-1} weighting so that for small motion and therefore for small temporal frequencies, the filters are particularly accurate. For temporal interpolators, the weighting function is zero in the region including spatial frequency π because the response of a symmetric even support filter must be zero at spatial frequency π [34]. The response of both filters is shown in Figure 5-5.

By collecting eight images in time, only the first three and a half harmonics can be examined. Higher harmonics are aliased to other frequencies. A straight forward analysis shows that energy in the higher harmonics will become significant when $A_x\omega_x + A_y\omega_y$ approaches 1, where A_i is the amplitude and ω_i is the highest spatial frequency in the i^{th} direction.

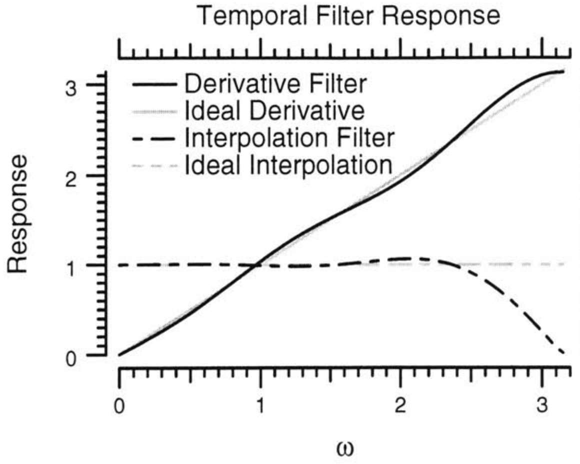

Figure 5-5: Frequency response of the 8-point temporal derivative and temporal interpolation filters.

5.4.2 Estimator Design

We seek to estimate the amplitude and phase of the sinusoidal motion based on eight velocity estimates. For simplicity, we illustrate the design process in one dimension. We assume motion in time of the form

$$d_x(t) = C_x \cos(\frac{\pi}{4}t) + S_x \sin(\frac{\pi}{4}t) \tag{5.4}$$

so that the velocity becomes

$$v_x(t) = -C_x\frac{\pi}{4}\sin(\frac{\pi}{4}t) + S_x\frac{\pi}{4}\cos(\frac{\pi}{4}t). \tag{5.5}$$

The goal is to determine (C_x, S_x) and use them to estimate amplitude and phase.

We consider two estimator designs. Using Equation (5.1), we obtain eight velocity estimates, one for each pair of adjacent images. We obtain an estimate of amplitude and phase using a discrete Fourier transform. To determine the displacement amplitude and phase, divide the velocity amplitude by the driving frequency and shift the phase by 90 degrees. We call this technique the DFT method. The DFT method yields the amplitude and phase of the first, second and third temporal harmonics.

A different technique uses the sinusoidal assumption directly. We make a linear least square (LLS) fit to velocity. We minimize

$$\operatorname*{arg\,min}_{C_x,S_x} \sum_{j,i,k} \left| \mathrm{G}_x[i,j,k]v_x(t) + \mathrm{G}_t[i,j,k] \right|^2 \tag{5.6}$$

from which we find

$$\begin{pmatrix} C_x \\ S_x \end{pmatrix} = \frac{1}{\frac{\pi}{4}} \begin{pmatrix} \sum \mathrm{G}_x^2 \sin^2(\frac{\pi}{4}k) & -\sum \mathrm{G}_x^2 \cos(\frac{\pi}{4}k)\sin(\frac{\pi}{4}k) \\ -\sum \mathrm{G}_x^2 \sin(\frac{\pi}{4}k)\cos(\frac{\pi}{4}k) & \sum \mathrm{G}_x^2 \cos^2(\frac{\pi}{4}k) \end{pmatrix}^{-1} \begin{pmatrix} \sum \mathrm{G}_x \mathrm{G}_t \sin(\frac{\pi}{4}k) \\ -\sum \mathrm{G}_x \mathrm{G}_t \cos(\frac{\pi}{4}k) \end{pmatrix} \tag{5.7}$$

where the sums are over i, j and k.

In the presence of noise, one might suspect that the LLS estimate would outperform the DFT method. However, this is not necessarily the case since each estimate of velocity uses the same data so that the measurements of velocity are not independent as the LLS method assumes. There is no guarantee that the LLS method will outperform the DFT method.

All of our simulations, both in the presence and absence of noise, have shown that the two methods produce insignificantly different results. For the rest of this document, we present the results using the LLS method only.

5.4.3 Evaluation for Images of Spatial Frequency Pairs

As a simple test image, we consider an image of the form $\cos(\omega_x i + \phi_x)\cos(\omega_y j + \phi_y)$. We choose the highest frequencies in the range of interest, $\omega_x = 2$, and $\omega_y = 2$ where the filter errors should be maximal. Results for $\phi_x = 0.5$ and $\phi_y = 2.34$ are presented in Figure 5-6 and Figure 5-7. We have chosen two algorithms. The first uses filters with 8x8x8 support. The second uses filters with 8x16x16 support. For small motions the algorithm that uses a larger support has smaller bias than than the algorithm that uses a smaller support. However, for motions larger than about 0.7 pixels, the two algorithms perform almost identically. Figure 5-7 shows the phase errors made by both algorithms. The errors made by each algorithm appear very similar. Examining the data, the phase errors made by the two algorithms are, in fact, identical.

The dependencies of errors on most factors other than amplitude are small. Specifically, changing the phases of the image ϕ_x and ϕ_y relative to the sampling positions has very little effect. Similarly, the dependence of the absolute phase of the motion has little difference. The relative phase of the motion in the x- and y- directions does make a small difference. The contours in Figure 5-6 and Figure 5-7 bulge by roughly 0.1 pixels (the performance improves slightly) when $v_x \approx v_y$ and the motions in each direction are 90 degrees of phase. We tested a different algorithm that used filters with support 1x17x8, 17x1x8 and 1x1x8 to estimate x, y, and t derivatives respectively; we also examined an algorithm that used 1x33x8, 33x1x8, and 1x1x8 filters. These algorithms gave nearly identical results to the 16x16x8 support algorithm.

We repeated the test of the four algorithms with $\omega_x = 1$, and $\omega_y = 1$. The simulated bias results were almost identical to the the biases shown in Figure 5-6 and Figure 5-7 except each of the axes

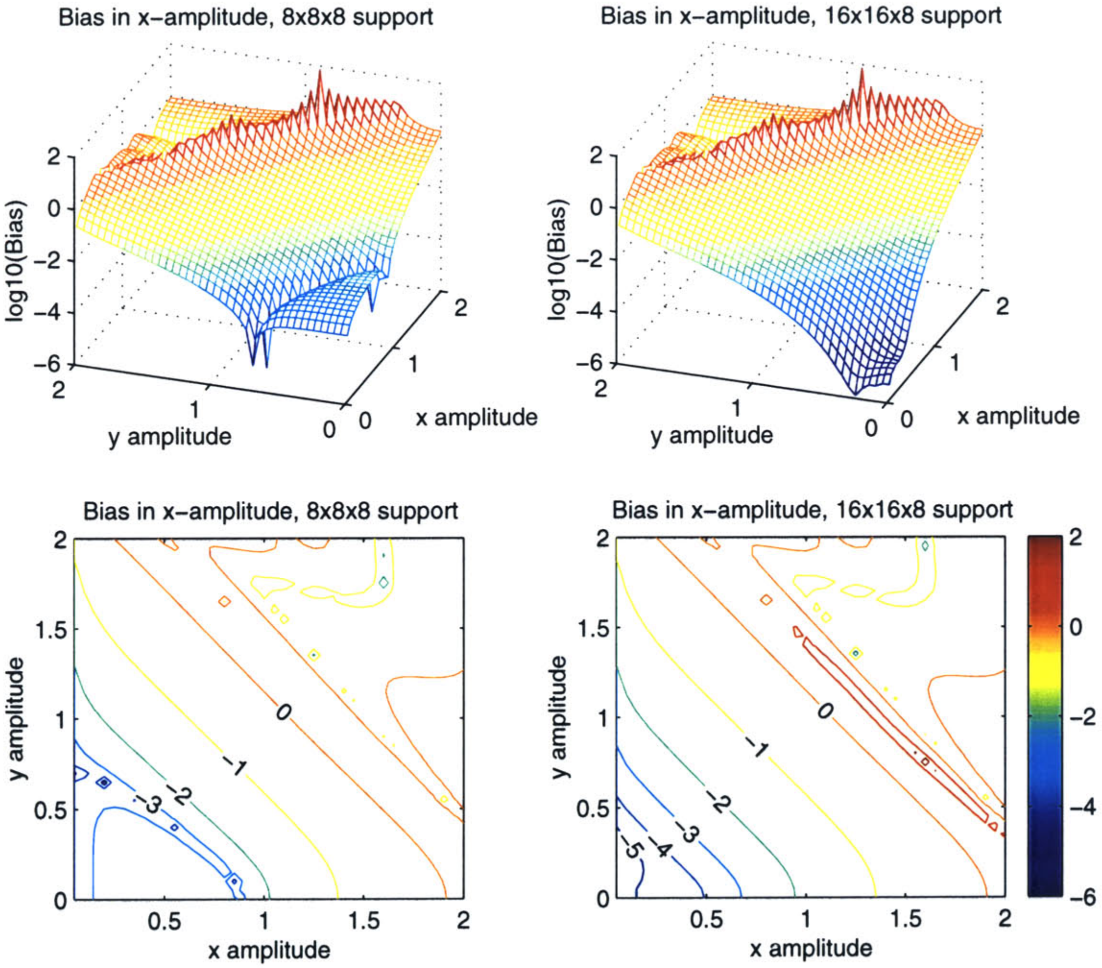

Figure 5-6: Error in the estimate of amplitude using eight images in time and either 8x8 or 16x16 support filters in space. The translating image consists of $\cos(2i + 0.5)\cos(2j + 2.34)$. The analysis region was 60 by 60 pixels. Motion in the x-direction was $A_x \sin(\frac{\pi}{4}t + 0.10)$; motion in the y-direction was $A_y \sin(\frac{\pi}{4}t + 0.8853975)$ where (A_x, A_y) are the tested amplitudes. All scales are in pixels.

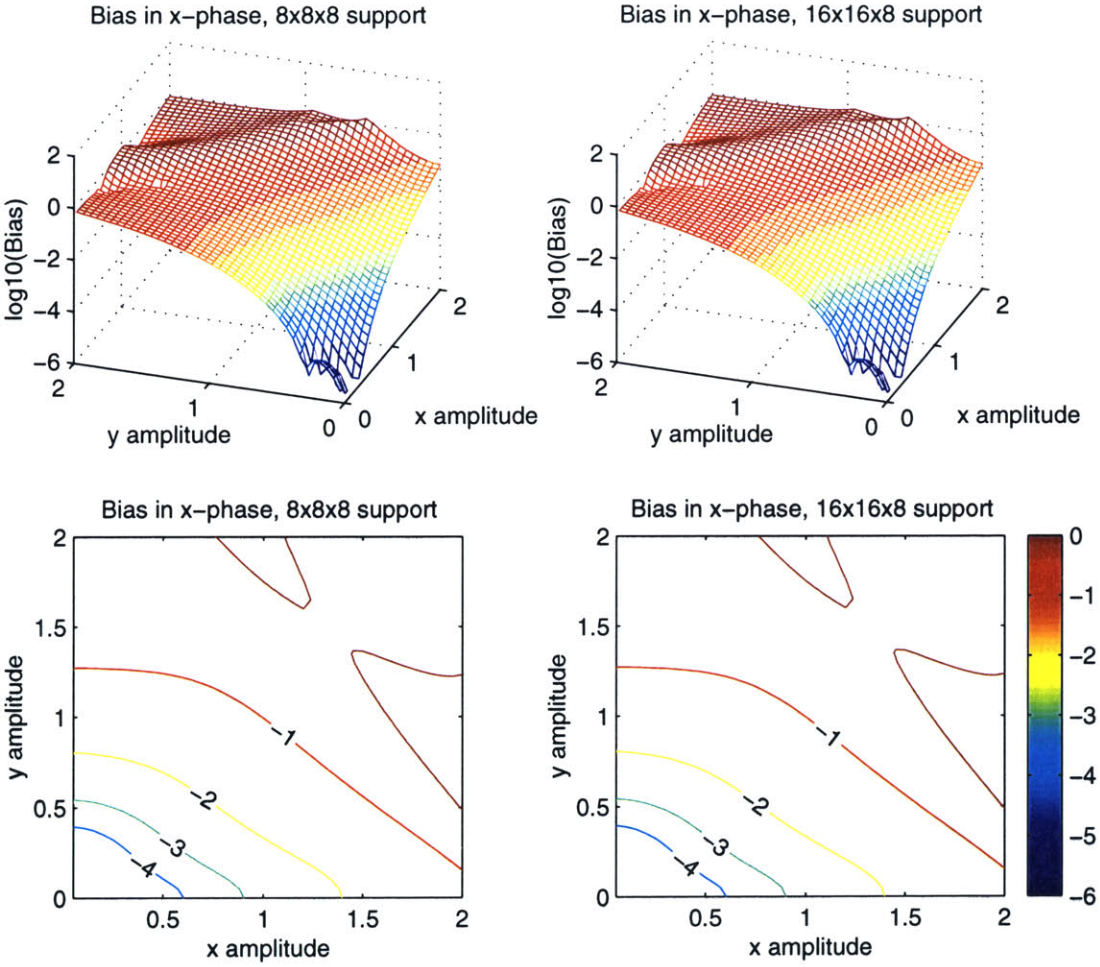

Figure 5-7: Error in the estimate of phase using eight images in time and either 8x8 or 16x16 support filters in space. The simulation is described in Figure 5-6. Phase bias is measured in radians.

scaled by a factor of 2.

5.4.4 Duty-Cycle Compensation

We call the fraction of the sinusoidal cycle during which an image is collected, the duty-cycle. A non-zero duty-cycle leads to blurring of moving objects. This effect can be modeled as a low-pass temporal filter. We create temporal derivative and interpolating filters that counteract the effects of blurring.

Consider the case of eight evenly spaced images per cycle with a $\frac{1}{8}$ duty-cycle. Uniform light intensity during the image acquisition time corresponds to the normalized low-pass filter $\frac{2\sin(\omega/2)}{\omega}$. We redesign our temporal filters to compensate for this effect. We call this duty-cycle compensation (DCC). The desired temporal derivative filter becomes $\frac{w^2}{2\sin(\omega/2)}$ and the desired temporal interpolating filter becomes $\frac{w}{2\sin(\omega/2)}$. We use the Parks-McClellan algorithm to create the filters using the weighting described in Section 5.4.1.

We repeat the simulations in the last section with a $\frac{1}{8}$ duty-cycle. In Figure 5-8 we compare the errors of a gradient-based algorithm using a 16x16x8 support designed with or without duty-cycle compensation. For small motions, the DCC algorithm yields amplitude errors that are 2 order of magnitude smaller than those of the algorithm without duty-cycle compensation. For larger motions, the estimators make similar amplitude errors. The phase errors of both algorithms are very similar at all amplitudes.

We tested the odd support filter algorithms described in the last section. Their performance was comparable to the 16x16x8 support filter algorithm, using duty-cycle compensation. Once again we found the errors of the 8x8x8 support algorithm were significantly greater than those of the 16x16x8 support algorithm.

5.4.5 The Addition of a Low-Pass Filter

We designed spatial filters that ignored frequencies above 2. The spectral content of images above this range will causes errors in derivative estimates. One way to reduce these errors is to low-pass filter images before the algorithms process them. Equivalently, we convolve all of the spatial filters in the algorithm with a low-pass filter. We have chosen a simple symmetric, 4 point filter to attenuate frequencies above 2. The response of the filter is plotted in Figure 5-9.

Adding a low-pass filter should have no effect on the results of the images just presented. The low-pass filter simply attenuates the amplitude of images of pairs of spatial frequencies. Simulations verified this assertion.

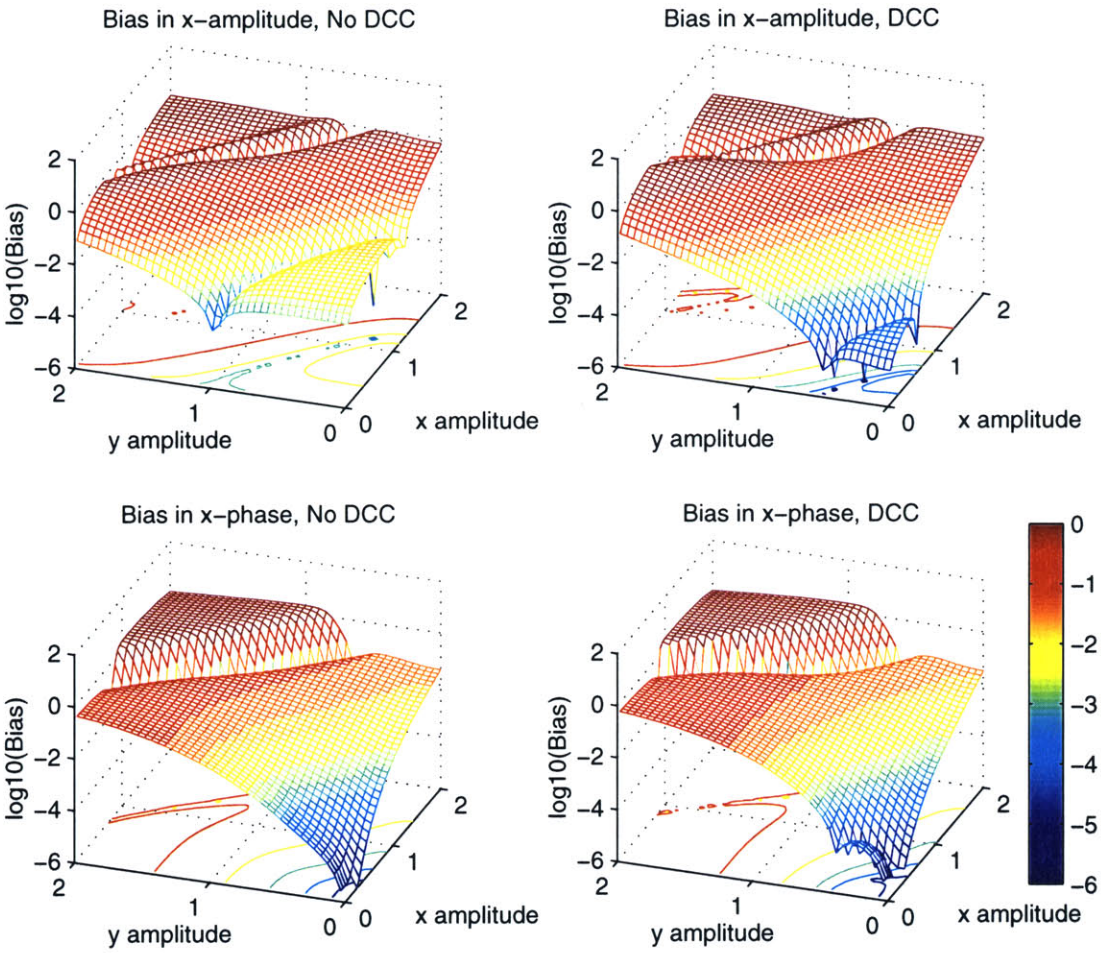

Figure 5-8: Error in the estimate of amplitude and phase for an multi-image gradient-based algorithm using a 16x6x8 support filter with or without duty-cycle compensation. The images are as described in Figure 5-6. All scales are in pixels with exception of the phase bias axis which is measured in radians.

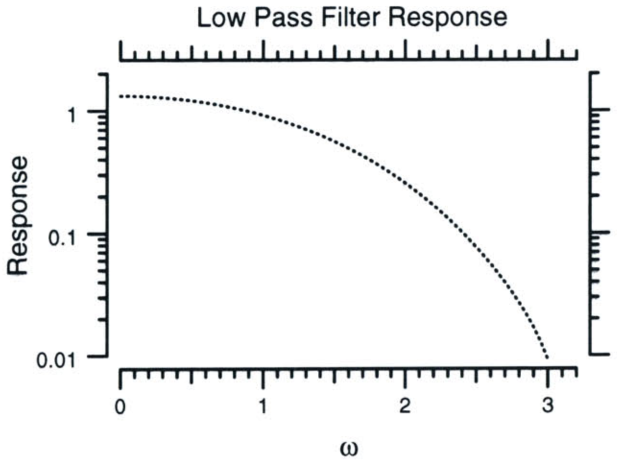

Figure 5-9: Frequency Response of the four point low-pass filter.

5.4.6 Discussion

As the motions increase in amplitude, the energy in aliased temporal harmonics increases. For large enough motions, we expect the errors in the temporal filters to dominate errors in the spatial filter and that due to those errors, the velocity estimator will yield poor estimates of motions. Figure 5-6 illustrates the effect. For small motion, $A_x\omega_x + A_y\omega_y << 1$, the temporal filters are almost exact so that errors in spatial filters dominate. Since the 8x16x16 support algorithm has exponentially smaller errors in it's spatial filters than the 8x8x8 support algorithm, for small motions it's errors are significantly smaller than those of the algorithm will less support. For larger motions, $A_x\omega_x + A_y\omega_y > 1$, the two algorithms perform almost identically, making errors that surpass one pixel. In this regime, the temporal filter errors dominate the spatial filter errors so that the two algorithms yield similar results.

It may seem surprising that in Figure 5-7 the two algorithms give identical errors in estimating phase. This result is limited to images of pairs of spatial frequencies. For this case, errors in a spatial filter lead to estimates of the velocity that are a constant multiplied by the actual velocity. Thus, the sine and cosine amplitudes both err by the same factor, leading to an accurate phase estimate.

That the phases of the sinusoidal motion, and the phases of the image led to small effects is not surprising. We believe that $A_x\omega_x + A_y\omega_y$ is the parameter of importance. This belief is supported by the finding that testing the same four algorithms with $\omega_x = 1$, and $\omega_y = 1$ yields almost identical bias plots to Figure 5-6 and Figure 5-7 except that each of the axes scaled by a factor of two.

Figure 5-8 shows that duty-cycle compensation is effective for reducing amplitude errors for small motions. However, for large motions, DCC has little effect. Duty-cycle compensation only effects the first three harmonics of the motion. Higher harmonics are aliased into other frequencies and therefore cannot be compensated for duty-cycle effects. Thus, DCC breaks down as motions become large enough so that the energy in higher harmonics becomes significant.

It is interesting to note that the estimates of phase for both algorithms in Figure 5-8 are almost identical. This stems from the fact that for images of pairs of spatial frequencies, the non DCC temporal filters yield a multiple of the correct answer for the sine and cosine amplitudes so that the phase estimator is unaffected. This does not hold for images with more complex spectra.

In the presence of noise, one must be careful using duty-cycle compensation. As the duty-cycle gets too large, the attenuation of temporal frequencies increases to the point that it becomes difficult to recover them from noise. Therefore, one must be careful not to use too large a duty-cycle.

5.5 Simulations of a Sinusoidal Motion Estimator

We simulate tests of the performance of the multi-image gradient-based sinusoidal motion estimators. We simulate sinusoidal motions of three test images, evaluating errors in the absence of noise to evaluate the inherent bias of the algorithms. We simulate sinusoidal motions of the same three test images in the presence of noise typical of optical systems. We compare the performance of LBC with that of the multi-image gradient-based estimators.

5.5.1 Methods

Algorithms

We tested the four multi-image gradient-based algorithms listed in table 5.1. For each multi-image algorithm, we used both the LLS and DFT methods to estimate amplitude and phase. Because the results of the eight different calculations were found to be similar, we report results using the LLS method with algorithm 16 . We also tested the pairwise linear bias compensated (LBC) image registration algorithm that was developed by Davis and Freeman [7] and then developed into a sinusoidal estimator [8].

Name	x-Derivative Filter Support	y-Derivative Filter Support	t-Derivative Filter Support
algorithm 14	11x11x8	11x11x8	11x11x8
algorithm 16	19x19x8	19x19x8	19x19x8
algorithm 15	20x4x8	4x20x8	4x4x8
algorithm 17	36x4x8	4x36x8	4x4x8

Table 5.1: Filter support of derivative calculations for the multi-image algorithms. The x- and y- axes are aligned respectively along the bottom and side of each image. AxBxC refers to the dimensions, in pixels, of the box shaped filter support region, respectively along the x,y and t directions. The filters include duty-cycle compensation and a 4 point low-pass filter. For each algorithm, we use the LLS and DFT methods to estimate amplitude and phase. Thus, we test a total of eight multi-image motion estimators.

In all cases, simulated results showed similar results along the x- and y-axes. Therefore, All estimates are reported as amplitudes and phase along the x-axis only.

Images

We analyze the performance of three different test images. The first image is of a dark spot in the form of a Hanning window on a bright background. This image was found to well simulate a spot used as experimental tests of the algorithms. The Hanning window took the form

$$G[i,j] = \begin{cases} M_0(1 - \frac{1+\cos 2\pi r/6}{7}) & \text{for } |r| < 3, \\ M_0 & \text{otherwise} \end{cases} \tag{5.8}$$

where r was the distance in pixels from the center of the spot and M_0 was adjusted to appropriately simulate the noise levels. To avoid artifacts that could result if the simulated spot were always aligned with the pixel grid, the center of the spot was chosen randomly, using a uniform distribution in a square bounded by the center of four pixels in a 2x2 box.

The second image is a polysilicon fatigue test structure designed by Exponent Failure Analysis Associates. The image of the fatigue test represents our interests in the motions of MEMS. The third images is a synthetic aperture radar (SAR) image of the Galapagos Islands[1].

Translations

Translations of the simulated spot were accomplished by recalculating images using Equation (5.8). Translations of the fatigue test device and SAR images were accomplished by taking the discrete Fourier transform of the 512x512 pixel images, multiplying by a pure phase and then inverse transforming the images. To avoid artifacts near the boundaries of the image, derivatives were evaluated in only the middle 32 by 32 pixel region. The filters used larger regions in support of their derivative estimates.

To simulate finite image acquisition time, we averaged 100 images evenly spaced in time during the acquisition period. For the fatigue test structure and SAR images, we transformed the images, averaged the results of a 100 different pure phase multiplications, and then inverse transformed the image.

For the test, motion in the x-direction was given by $A_x \sin(\frac{\pi}{4}t - 1.3)$ and motion it the y-direction given by $A_y \ \sin(\frac{\pi}{4}t - 1.4)$. For each test consisting of a 100 sequences of images, a different pair of amplitudes, A_x and A_y, were chosen.

Simulated Noise

We simulate the effects of noise introduced by a CCD camera, including fixed-pattern noise, shot noise and quantization errors. Motion was simulated by creating a sequence of eight high-precision images representing the average number of electrons collected at each pixel during the duty-cycle. To simulate fixed pattern noise, each pixel (i, j) was then multiplied by a random gain factor taken from a Gaussian distribution. The (i, j) pixel in each set of eight images was multiplied by the same gain factor. The standard deviation of the Gaussian distribution was set to 0.00315 so that it simulated -50 dB of fixed-pattern noise relative to the average signal energy in the image. Shot noise was simulated using a pseudo-random Poisson number generator at each pixel. The number of electrons was then divided by 32, the number of electrons needed to increment the A/D output of a camera by one, and then truncated. The gain was chosen to set the shot noise to be -50 dB relative to the average signal energy in the analysis region of each image.

[1] Image (P-43899) courtesy of the Jet Propulsion Laboratory, Pasadena, CA. http://www.jpl,nasa.gov/archive/mpe2.html.

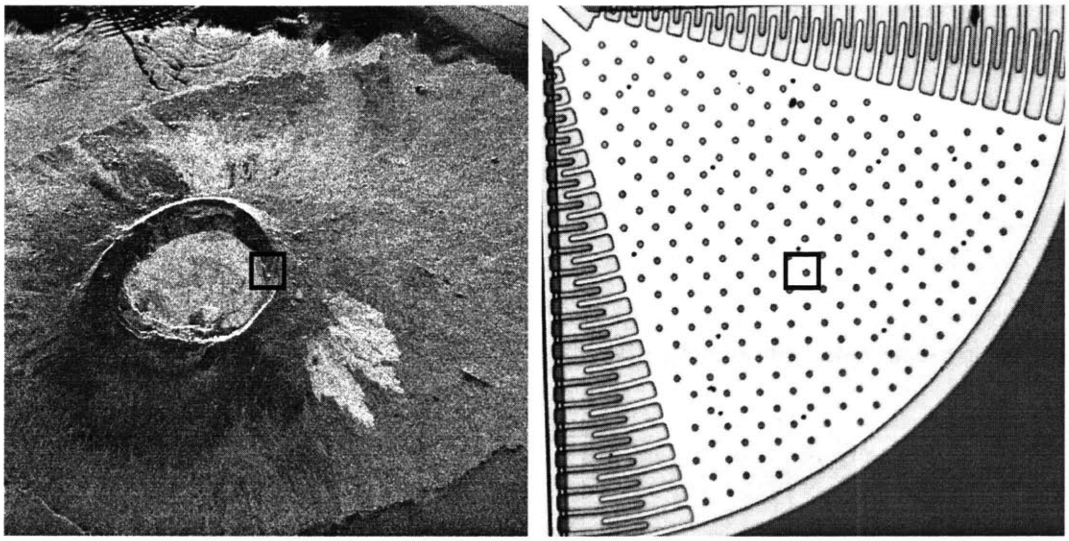

Figure 5-10: Test Images. On the left is an SAR image of the Galapagos Islands. On the right is the polysilicon failure device. The boxes indicate analysis regions.

All noise simulations were repeated using 100 sequences of images consisting of 10 different fixed-noise patterns. Each image in the 100 sequences received a different shot noise distribution.

5.5.2 Performance in the Absence of Noise

To investigate inherent bias in the multi-image gradient-based algorithms, we simulated their performance in the absence of noise. Figure 5-11 shows the performance of algorithm 16 for the SAR and fatigue test device images. There are a number of important similarities in the errors of algorithm 16 in both images. For both images, amplitude errors remained below 0.001 pixels for amplitudes smaller than 1.2 pixels. For larger motions, the amplitude errors gradually increase, passing 0.01 pixels. This trend of gradually increasing errors was also seen in Figure 5-6 for the image of the product of cosines. Error in amplitude estimates of the spot (not shown) were also less than 0.001 pixels for amplitudes less than 1.2 pixels. The same trend of gradually increasing error was also seen for the spot. Phase errors using the image of the SAR and Failure Device remain below 10^{-2} radians for all cases, dropping well below 10^{-4} radians for small motions. For the image of the bead (not shown), phase errors remained below 10^{-3} radians for all amplitudes examined.

5.5.3 Performance in the Presence of Noise

Figures 5-12 and 5-13 shows the effects of adding -50 dB of shot and fixed pattern noise to the spot, failure device and SAR image. For the spot, the amplitude bias in algorithm 16 is almost uniform remaining near 0.01 pixels. Conversely, the amplitude bias of LBC is slightly less than 0.1 pixels for most amplitudes. Standard deviations for both algorithms (not shown) were almost uniform at 0.01

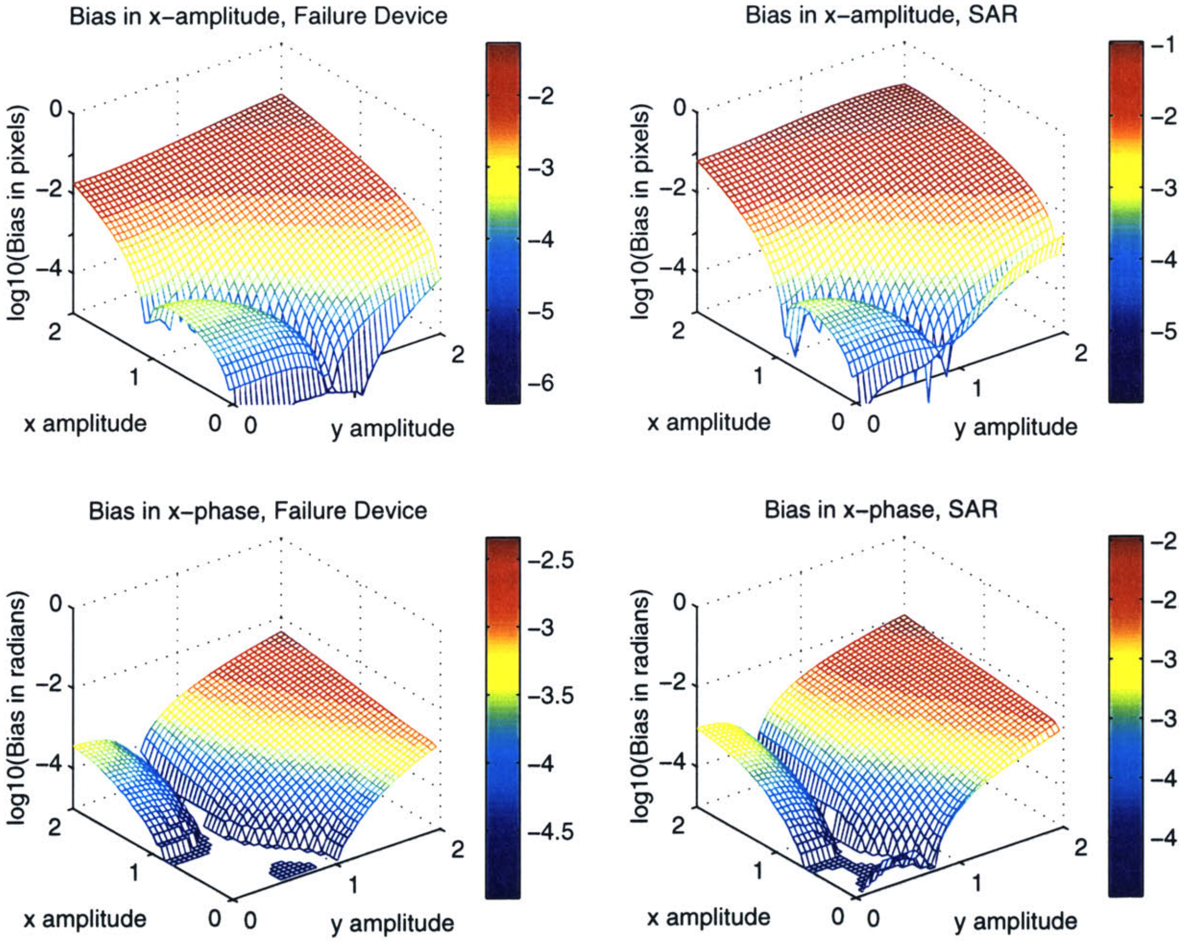

Figure 5-11: Error in the estimate of amplitude and phase using algorithm 16 . Each trial consisted of eight simulated images acquired with $\frac{1}{8}$ duty-cycle at evenly space phases of the sinusoidal motion. The simulation is described in detail in Section 5.5.1.

pixels.

For the SAR and fatigue test structures, the amplitude errors of algorithm 16 in the presence of noise are quantitatively similar to the amplitude errors in the absence of of noise shown in Figure 5-11. For motions smaller than roughly 1.2 pixels, amplitude errors are smaller than 0.001 pixels. Conversely, LBC's errors almost reach 0.1 pixels for the same amplitudes. However, for large enough motions, Algorithm 16's errors are larger than those of LBC. For both images, standard deviations were uniform at 0.001 pixels.

Phase errors for algorithm 16 for the image of the bead remain near 10^{-3} radians. Conversely, phase errors of LBC become almost as large as 10^{-2} radians. For both algorithm 16 and LBC, phase errors for the SAR image and fatigue test structure image are quantitatively similar. Algorithm 16's phase errors range from below 10^{-4} radians up to 10^{-2} radians. LBC's phase errors remain mostly between 10^{-3} and 10^{-2} radians. The standard deviation for estimates of phase by algorithm 16 and LBC for the image of the fatigue test device are shown in Figure 5-14. The standard deviations ranges from about 3×10^{-3} to 10^{-2} radians. The standard deviations of phase by algorithm 16 and LBC for the image of the SAR device were quantitatively similar to those of the SAR image. Standard deviations for phase estimates of the motion of the bead ranged from a factor of three to a factor of five higher for both algorithms.

Algorithm 15 and algorithm 17 performed as well as algorithm 16 using both the LLS and DFT methods. For all cases, similar results were obtained in for motion estimates along the y- and x-axes.

5.5.4 Discussion

Figure 5-11 shows that in the absence of noise, the inherent bias in the multi-image gradient-based algorithms is smaller than 0.001 pixels for small enough motions. However, for larger motions, the multi-image gradient-based algorithms make errors larger than 0.1 pixels. The multi-image gradient-based methods do not perform well for large motions. The difficulty, in the case of sinusoidal motion, is that the brightness of a pixel does not vary sinusoidally with time. For large motions the temporal frequencies in the brightness function become larger than π and are aliased into other temporal frequencies, causing errors.

In the presence of noise, the standard deviations of the multi-image based methods are comparable to those of LBC. For the SAR and fatigue test structure images, for motions smaller than 1.2 pixels, the amplitude error of the multi-image gradient methods is more than an order of magnitude smaller than that of LBC. However, for motions approaching 2 pixels, the amplitude errors of the multi-image gradient methods exceed those of LBC. Similarly, algorithm 16 makes significantly s-maller amplitude errors for small enough motions. For motions approaching two pixels in amplitude in either direction, the phase errors in both algorithms become similar.

We conclude that the multi-image based algorithms perform significantly better than LBC for

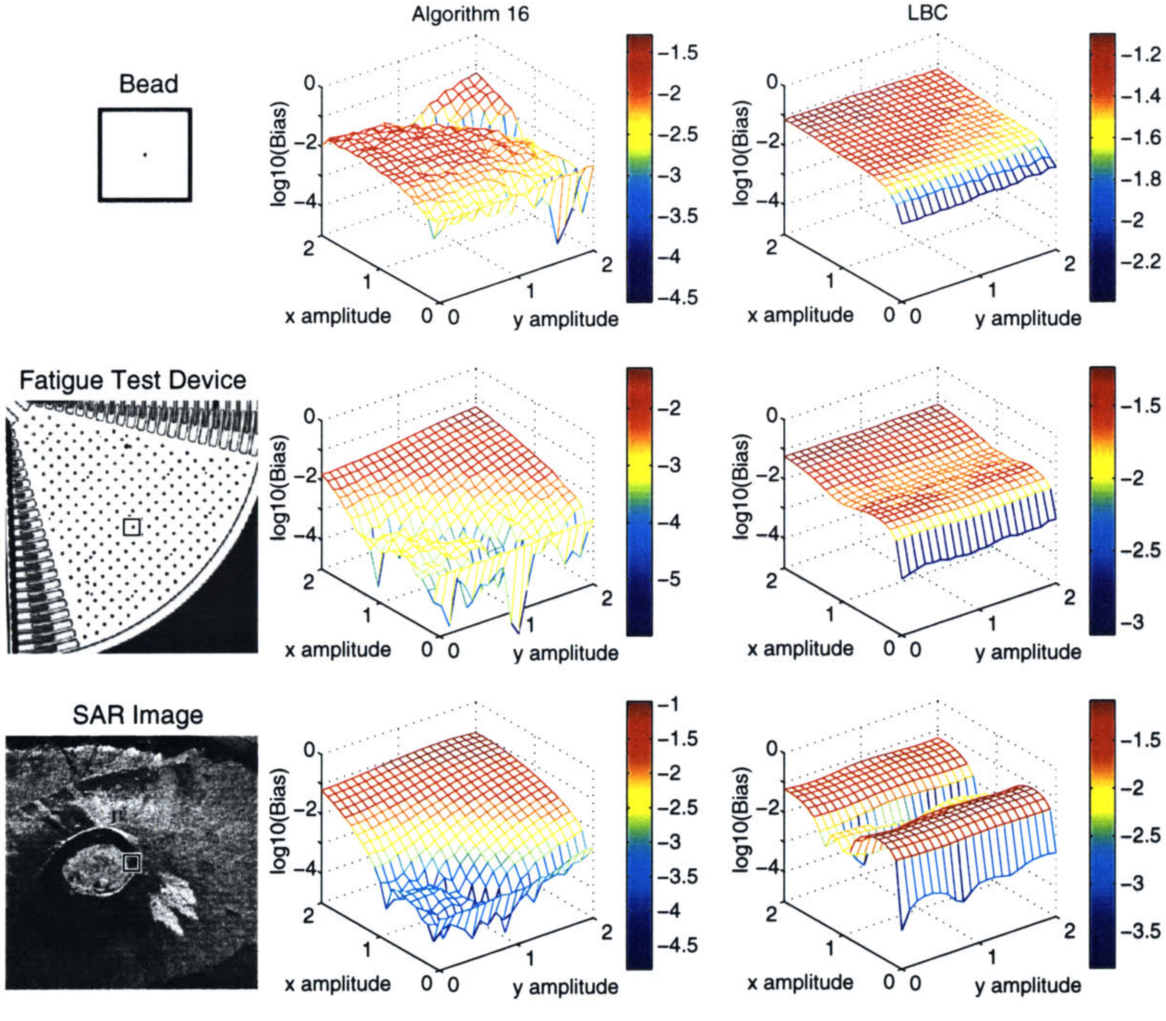

Figure 5-12: Bias for the estimate of amplitude using algorithm 16 (left) and LBC (right) in the presence of -50 dB of shot noise and fixed-pattern noise. All axes are in pixels. The simulation are described in Section 5.5.1 and Section 5.5.1

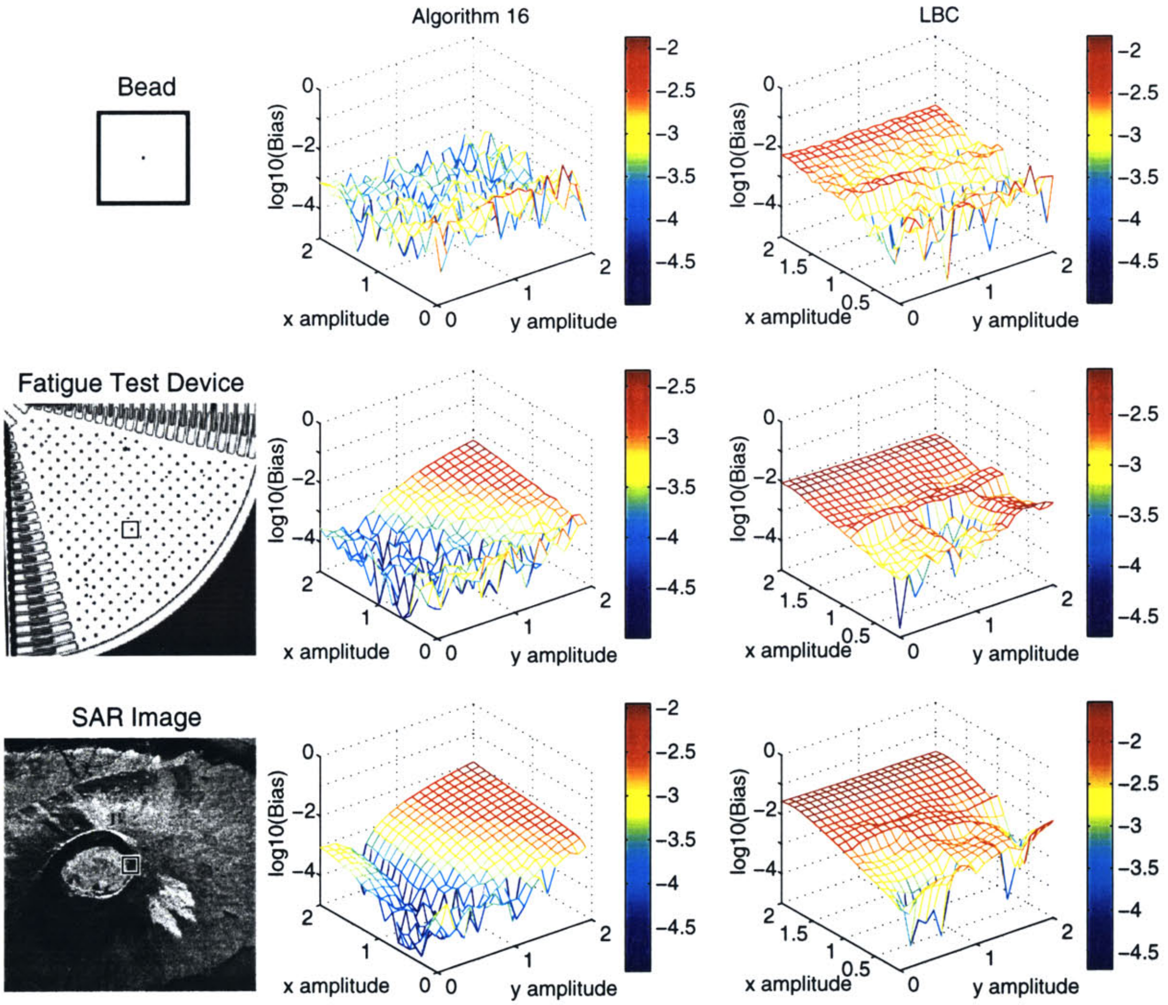

Figure 5-13: Bias for the estimate of phase using algorithm 16 (left) and LBC (right) in the presence of -50 dB of shot noise and fixed-pattern noise. Phase is measured in radians. The simulation are described in Section 5.5.1 and Section 5.5.1

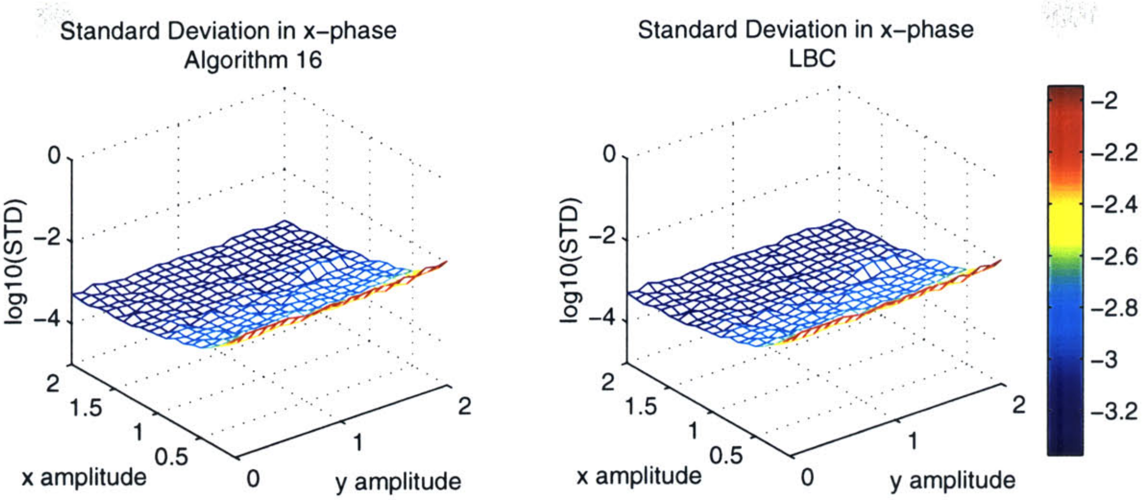

Figure 5-14: The standard deviation of phase errors for algorithm 16 for simulations of the fatigue test structure image. The simulations are described in Section 5.5.1 and Section 5.5.1

motions smaller than about 1.2 pixels, and comparably to LBC for motions up to 2 pixels. This gain in performance comes at computational costs. While the number of computations for both algorithms scales linearly with the number of pixels in an analysis region, multi-image gradient-based algorithms require between tens and thousands more multiplications and additions per pixel in the analysis region.

It is possible to improve the dynamic range of the multi-image based algorithms. Either by down-sampling the images, making super-pixel motions into sub-pixel motions or by equivalently low-pass filtering an image, the multi-image gradient-based methods can be made to work for larger motions.

5.6 Conclusions

We have developed a method based on cascades of filters to create multi-image gradient-based algorithms. We have illustrated this technique for sinusoidal motion where we have shown that it is possible to achieve errors inherent in the algorithm smaller than 0.001 pixels in amplitude and smaller than 0.001 radians in phase for motions smaller than 1.2 pixels. We have shown that in the presence of -50 dB shot noise and fixed-pattern noise, the algorithms still achieve error this small.

We have shown that for larger motions, the errors inherent to the algorithm increase, eventually becoming comparable to existing algorithms such as LBC. However, by down-sampling or low-pass filtering an image, it is possible to increase the dynamic range of the method.

While we have illustrated the use of the multi-image technique for sinusoidal motion, the same methods are applicable for arbitrary motion. The limitations of the method are that images must be sampled frequently enough in time to satisfy the Nyquist criterion.

5.7 Filters

Length	Temporal Filters		
Interp., no DCC	−0.0249546183106627	0.0835526882564852	−0.187030489727413
	0.628370372733361	0.628370372733361	−0.187030489727413
	0.0835526882564852	−0.0249546183106627	
Deriv., no DCC	−0.0489387545422273	0.0696364235870484	−0.158276184031824
	1.28910639031026	−1.28910639031026	0.158276184031824
	−0.0696364235870484	0.0489387545422273	
Interp., DCC	−0.0378010678346327	0.125047021427472	−0.267629124130556
	0.680287727944692	0.680287727944692	−0.267629124130556
	0.125047021427472	−0.0378010678346327	
Deriv., DCC	−0.14903590789871	0.204171130411121	−0.408622311811501
	1.69565453432943	−1.69565453432943	0.408622311811501
	−0.204171130411121	0.14903590789871	

Length	Spatial Derivative Filter		
8	−0.00286216383353546	0.0194079387912599	−0.102277846435357
	1.22945478993064	−1.22945478993064	0.102277846435357
	−0.0194079387912599	0.00286216383353546	
16	−2.7006484352869e−05	0.000214437570001811	−0.000983311903970464
	0.00347041534087513	−0.0107467889783358	0.0328020859998998
	−0.121030730300938	1.25150829022001	−1.25150829022001
	0.121030730300938	−0.0328020859998998	0.0107467889783358
	−0.00347041534087513	0.000983311903970464	−0.000214437570001811
	2.7006484352869e−05		
17	−0.000502321109685936	0.00336800530337642	−0.0128450407337949
	0.0367375494850381	−0.0877302533889347	0.187633106381368
	−0.388930509333614	0.939689807053354	0
	−0.939689807053354	0.388930509333614	−0.187633106381368
	0.0877302533889347	−0.0367375494850381	0.0128450407337949
	−0.00336800530337642	0.000502321109685936	
33	−3.49589617109824e−07	3.94225697555072e−06	−2.41437567272228e−05
	0.000105884231519362	−0.000370560391102016	0.0010956938320803
	−0.0028351294807867	0.00657523476932475	−0.0139135332551632
	0.0272544346753638	−0.0500738402018483	0.0874842787085976
	−0.147882293734463	0.24853122026952	−0.439078709811418
	0.968118055508647	0	−0.968118055508647
	0.439078709811418	−0.24853122026952	0.147882293734463
	−0.0874842787085976	0.0500738402018483	−0.0272544346753638
	0.0139135332551632	−0.00657523476932475	0.0028351294807867
	−0.0010956938320803	0.000370560391102016	−0.000105884231519362
	2.41437567272228e−05	−3.94225697555072e−06	3.49589617109824e−07
Length	Spatial Interpolating Filter		
8	−0.0122099803061808	0.0538078242669685	−0.158462745547521
	0.616840504977136	0.616840504977136	−0.158462745547521
	0.0538078242669685	−0.0122099803061808	
16	−0.000255171473531746	0.0016387438183934	−0.00602175636195368
	0.0167166926834678	−0.0391305570001712	0.0836286567793996
	−0.182808582018383	0.626231904061036	0.626231904061036
	−0.182808582018383	0.0836286567793996	−0.0391305570001712
	0.0167166926834678	−0.00602175636195368	0.0016387438183934
	−0.000255171473531746		
Length	Spatial Low Pass Filter		
4	0.14962930880927	0.51071846633449	0.51071846633449
	0.14962930880927		

Chapter 6

Development and Simulation of Fourier Transform Based Image Registration Techniques

6.1 Abstract

We develop a two-image Fourier transform based motion estimator. For comparison with previous chapters, we build on this algorithm to make an estimator of amplitude and phase of sinusoidal motion. In the presence of noise typical for our applications, −50 dB of fixed-pattern noise and −50 dB of shot noise, amplitude errors of 0.1 pixels and phase errors of 0.01 radians are achievable. However, these errors are larger than those of already existing algorithms.

6.2 Introduction

Gradient-based methods rely on derivatives in space and time to estimate motion. While these methods are effective, they are limited both by the order to which derivatives are estimated and by accuracy limitations in calculating the derivatives.

A different approach to image registration is the maximization of the output of a statistical correlator. For subpixel motion estimation, such a technique is limited by its interpolation method and by the ability of a search algorithm to find the global maximum in the correlator. Since it is difficult to guarantee that a search will find the maximum, one would like to find a correlation-based method that does not require a search. Fourier transform based correlation methods satisfy this criteria. A translation in space between two images causes a phase shift in the Fourier domain.

To detect motions, we develop a least squares estimator to detect phase differences in the Fourier domain between two images.

We make an estimator of amplitude and phase of sinusoidal motion based on a two-image registration algorithm. We simulate the performance of this estimator in the presence of a duty cycle and noise as we did in Chapter 5.

6.3 Development of the Estimator

We start by developing the theory of Fourier-based correlation methods. We choose a window function to use with the estimator. Then, we examine which frequencies to examine in the estimator. To implement the pairwise registration algorithm, we develop a phase unwrapping technique. We finish this section by developing an estimator of amplitude and phase for sinusoidal motion.

6.3.1 Theory of Fourier-Based Correlation

Knapp and Carter [22] review Fourier-based methods. We closely follow their derivation. For simplicity, we consider a one dimensional signal corrupted by additive, zero mean, stationary random noise. We make the additive noise approximation more because it yields mathematically tractable results than because it realistically describes our data[1].

Consider $f(x)$, a band-limited continuous signal in space, and a shifted version $f(x+d)$, both of which are subject to additive noise. A detector measures

$$y_1(x) = f(x) + n_1(x), \tag{6.1}$$

$$y_2(x) = f(x+d) + n_2(x) \tag{6.2}$$

where $y_1(x)$ and $y_2(x)$ are the detected signals and $n_1(x)$ and $n_2(x)$ are additive noise sources with identical power spectrums, independent of each other and independent of the data. The cross correlation function is given by

$$R_{y_1,y_2}(\delta) = E[y_1(x)y_2(x-\delta)] = \int_{-\infty}^{\infty} Y_1(\omega)Y_2^*(\omega)e^{j\omega\delta}d\omega \tag{6.3}$$

where E denotes expectation, Y_1 is the Fourier transform of y_1 and Y_2^* is the complex conjugate of the Fourier transform of y_2. The cross correlation is a real function with a maximum near $\delta = d$. The detector samples the functions y_1 and y_2 at a rate that is assumed to be greater than Nyquist's

[1] CCD signals are corrupted mainly by shot noise, quantization noise, fixed-pattern noise and read noise [21]. The approximation to additive noise may be satisfactory if shot noise dominates other noises and only small intensity variations exist spatially.

frequency. The correlator becomes

$$R_{y_1,y_2}(\delta) = \int_{-\pi}^{\pi} Y_1(\Omega)Y_2^*(\Omega)e^{j\Omega\delta}d\Omega \tag{6.4}$$

where δ is now measured in pixels. The detector samples a rectangular window of data. Is it often beneficial to multiply the data by a particular window function because estimates of spectrums are convolved with the Fourier transform of a window function [34]. We discuss the choice of window function to the next section.

Since the signal to noise ratios of Y_1 and Y_2 will vary as a function of frequency, it is typically beneficial to filter y_1 and y_2 before finding the cross correlation function. Knapp and Carter review several filters. We use the filter that achieves the Cramér-Rao bound. After normalizing the cross spectral density, the appropriate frequency weighting simplifies to the signal to noise ratio, $\frac{P(\Omega)}{N(\Omega)}$, where $P(\Omega)$ is the signal power and $N(\Omega)$ is the noise power. The correlator becomes

$$R(\delta) = \int_{-\pi}^{\pi} \frac{P(\Omega)}{N(\Omega)} \frac{Y_1(\Omega)Y_2^*(\Omega)}{|Y_1(\Omega)Y_2(\Omega)|}e^{j\Omega\delta}d\Omega = \int_{-\pi}^{\pi} \frac{P(\Omega)}{N(\Omega)}e^{j[\phi(\Omega)-\Omega\delta]}d\Omega \tag{6.5}$$

where we have replaced the normalized cross spectral density by a phase $e^{j\phi(\Omega)}$.

The right side of Equation (6.5) is maximized by the phase-plane $e^{j\Omega\delta}$ that best correlates with the filtered cross spectral density function of the two images. We avoid a search by fitting a plane to the phases of the discrete Fourier frequencies using a least squares approach. To accomplish this, remember that the cross correlator is real so that one need only consider the real portion of the complex exponential, a cosine. Under high signal to noise conditions, we expect $\phi(\Omega) \approx \Omega d$ so that the maximum of $R(\delta)$ will occur close to d. Therefore, $\phi(\Omega) - \delta$ will be small near the maximum so that one can expand the cosine and keep only the lowest order terms that depend on $\phi(\Omega) - \Omega\delta$. Since $\cos(x) = 1 - \frac{x^2}{2} + O(x^4)$, we find

$$R(\delta) \approx \int_{-\pi}^{\pi} \frac{P(\Omega)}{N(\Omega)} \left[1 - \frac{1}{2}(\phi(\Omega) - \Omega\delta)^2\right] d\Omega. \tag{6.6}$$

We approximate the integral by evaluating the integrand at the discrete Fourier transform frequencies. We ignore terms that have no dependence on δ so that maximizing R is equivalent to

$$\widehat{\delta} = \arg\min_{\delta} \sum_{\Omega=\frac{2\pi}{M},\frac{4\pi}{M},\dots,\pi} \frac{P(\Omega)}{N(\Omega)}(\phi(\Omega) - \Omega\delta)^2 \tag{6.7}$$

where M is the number of points in the analysis region, and $\widehat{\delta}$ is the motion estimate. We have excluded the contributions of half the discrete Fourier spatial frequencies due to symmetries of the discrete Fourier transform [34].

In two dimensions, a similar derivation of the estimator leads to

$$(\widehat{\delta_x}, \widehat{\delta_y}) = \arg\min_{\delta_x, \delta_y} \sum_{\substack{\Omega_y = -\pi, \dots, \pi \\ \Omega_x = \frac{2\pi}{M}, \frac{4\pi}{M}, \dots, \pi}} \frac{P(\Omega_x, \Omega_y)}{N(\Omega_x, \Omega_y)} (\phi(\Omega_x, \Omega_y) - \Omega_x \delta_x - \Omega_y \delta_y)^2 \tag{6.8}$$

where $(\widehat{\delta_x}, \widehat{\delta_y})$ is the motion estimate and (Ω_x, Ω_y) is the discrete Fourier spatial frequency. We have summed the contributions due to half the discrete Fourier spatial frequencies rather than all of them due to symmetries in the two-dimensional discrete Fourier transform [27]. Minimizing, we find

$$\begin{pmatrix} \widehat{\delta_x} \\ \widehat{\delta_y} \end{pmatrix} = \begin{pmatrix} \sum \frac{P}{N} \Omega_x^2 & \sum \frac{P}{N} \Omega_x \Omega_y \\ \sum \frac{P}{N} \Omega_x \Omega_y & \sum \frac{P}{N} \Omega_y^2 \end{pmatrix}^{-1} \begin{pmatrix} \sum \frac{P}{N} \phi \Omega_x \\ \sum \frac{P}{N} \phi \Omega_y \end{pmatrix} \tag{6.9}$$

where the sums are over the discrete Fourier frequency pairs Ω_x, Ω_y.

6.3.2 Choice of Window Function

Window functions average spectra in the frequency domain. The amplitude and phase estimated at each discrete Fourier frequency (Ω_x, Ω_y) are the weighted averages of the amplitudes and phases of a band of frequencies centered at (Ω_x, Ω_y). If the spectral energy in the image varies over a band, then the weighted average introduces bias into the estimator. To see this, consider that our derivation of (6.9) attributes all the energy in a band to the center frequency in the band. If, for example, the lower half of the band has most of the spectral energy in the band, then attributing all the energy to the center frequency causes errors in the estimator. Since the bands become smaller as the analysis region increases in size, the spectral energy in a band will become more uniform as the analysis region increases and the bias in a Fourier transform based estimator will decrease. In fact, Fourier-based methods are asymptotically unbiased as the analysis region becomes large [40].

Window functions have side lobes which average frequencies outside the main band. These side lobes also create bias in the estimates of amplitude and phase. We seek a tradeoff between side lobe amplitude and main lobe width. A table of common window functions is found in [34]. We examined the rectangular window function because it has the main lobe with the smallest width. It has side lobes with magnitude −13 dB relative to the magnitude of the main lobe. The large relative height of the side lobes generally causes the rectangular window function to perform poorly in the presence of noise. By doubling the allowable width of the main lobe, we can choose from from a variety of window functions with much smaller side lobes. We examine the Hanning window based estimator whose maximum side lobe amplitude is −31 dB. For our tests in the presence of noise, the Hanning window always outperformed the rectangular window based estimator. Therefore, we report results exclusively for the Hanning window.

6.3.3 Discrete Fourier Frequency Exclusion

The discrete Fourier transform of an image is equivalent to the Fourier transform of the image tiled infinitely in all directions [34]. At the edges of the image mismatches occur: for example, the right edge of one image abuts the left edge of a tiled image. These mismatches cause high frequencies which, through the sidelobes of the window function, causes bias in the estimators of amplitude and phase at a given discrete Fourier frequency pair. The errors are particularly prevalent along the $\Omega_x = 0$ and $\Omega_y = 0$ axes. We therefore eliminated all discrete Fourier frequency pairs from the sums in (6.9) for which either of the frequencies are 0.

Willsky et.al [40] calculates the Cramér-Rao bound for estimating phase using a rectangular window function. For frequencies within $\frac{\pi}{M}$ of 0 or π, the minimum variance of the phase estimate is highly dependent on the phase, potentially becoming infinite. Similar results apply to the phase of a cross-spectral density. For a Hanning window, the same results apply except the range of frequencies changes to within $\frac{2\pi}{M}$ of 0 or π.

We do not wish to include frequencies in the sums in (6.9) for which the minimum variance of the phase estimate may be infinite. For the rectangular window function, we therefore do not include any frequencies pairs in which either frequency is 0 or π. For the Hanning window function, we do not include any frequency pairs in which either frequency is 0,π,$\frac{2\pi}{M}$, or $\pi - \frac{2\pi}{M}$. If the images to be registered have a large D.C. component, the Hanning windowed Fourier transform will have a large amplitude at frequency $\frac{2\pi}{M}$ whose phase will be independent of the motion. Therefore, it is reasonable to ignore this frequency in the sum.

6.3.4 Implementation

To implement Fourier transform based registration, we must estimate the weighting function $\frac{P}{N}$ as a function of frequency. Based on the data in Chapter 2, we note that fixed-pattern noise, the dominant source of noise, is roughly independent of frequency. We assume that the noise is independent of frequency so that N is reduced to a scale factor. To estimate a power spectrum, we use the magnitude squared of the Fourier transform of the windowed image. When estimating the pairwise-motion between two images in a sequence, we use the average of the power spectrums of each of the images in the sequence as the weighting function.

One difficulty with Fourier-based methods is that phase is defined only in the range from $-\pi$ to π. When ϕ becomes larger than π, it's value wraps to $-\pi$; this effect is known as phase wrapping. In the absence of noise, phase wrapping occurs in the Fourier domain in the region $|\Omega_x\delta_x + \Omega_y\delta_y| > \pi$. We address phase wrapping by noting that if both δ_x and δ_y are known to be smaller than a certain value, there is a diamond shaped region in the Fourier domain in which phase wrapping will not occur. This region includes the lowest spatial frequencies where, for our applications, the highest signal to noise ratios exist. Thus, we should be able to obtain a good estimate of the motion from the diamond shape region. Once the estimate of motion exists, we subtract the phase due to the motion estimates to obtain a residual phase. If the initial motion estimates in each direction are within 0.5 pixels of the actual motions, no phase wrapping will occur in the residual phase. We can then use the entire region to estimate a correction to the original motion estimate. We assume that δ_x and δ_y are known to within 2 pixels. This statement could hold either because the motion is known to be small or because a different algorithm was used to estimate the motion to within 2 pixels.

In practice, rather than using two passes to estimate the displacements, we iterate on the last step so that residual phases which are close to π and $-\pi$ are phase unwrapped correctly. For the cases we have examined, the algorithm has not needed more than the initial motion estimate using the diamond shape region and two iterations. We stop the iteration process after the motion estimate ceases to change by more than 10^{-6} pixels or after 100 iterations.

We summarize the two-image registration algorithm:

1. Estimate power spectrum of the signal $P(\Omega_x, \Omega_y)$.
2. Find an estimate of δ_x and δ_y to within 2 pixels.
3. Subtract the phase due to the motion estimate from the cross spectral phase of the two images.
4. Estimate the motion using equation (6.9), summing over frequencies in a diamond shape region as described in the text.
5. Subtract the phase due to the estimate from the remaining cross spectral phase.
6. Estimates the motion using the entire frequency domain.
7. Repeat steps 5 and 6 until the new estimate ceases to change by more than 10^{-6} pixels, or after 100 iterations.

6.3.5 Sinusoidal Estimation

We design a sinusoidal estimator assuming eight evenly spaced images during a cycle as in Chapter 5. We calculate seven pairwise motions between images 1 and 2, 2 and 3, ... , and 7 and 8. We use the motion estimates to calculate position versus time, defining the position at the first image to be 0. We then use an 8 point discrete Fourier transform to estimate amplitude and phase.

If images are acquired using a $\frac{1}{8}$ duty cycle, moving objects are blurred in the acquired image. It is possible to compensate the images for blurring before calculating the pairwise motions. We do this using the 8 point duty-cycle compensated (DCC) temporal interpolation filter found in Section 5.7. We then use the eight compensated images to estimate amplitude and phase. The interpolator has a time lag of $\frac{1}{2}$ temporal unit which causes of a phase shift of $\frac{\pi}{8}$. We adjust the phase estimate to compensate for the phase lag of the interpolator.

6.4 Methods

Our methods are almost identical to those of Chapter 5. As in Chapter 5, we test the synthetic aperture radar (SAR) image of the Galapagos Islands, the image of the fatigue test device and the image of the spot. We also test a simulated image of a rod and a spot.

As in Chapter 5, for each simulation of one pair of amplitudes, the center of the spot was chosen randomly, using a uniform distribution in a square bounded by the center of four pixels in a 2x2 box.

As in Chapter 5, we use Fourier transform based methods to shift the image of the SAR and fatigue test device images. To minimize the effects of using a discrete Fourier transform to simulate motions and another transform to detect motions, we use a 512x512 pixel support for the shifting transform, much larger than the registration transform.

We consider analysis region sizes of 32 by 32 and 64 by 64 pixels. We examine displacement amplitudes between 0 and 2 pixels in both directions. We choose not to consider larger motions so that the phase wrapping algorithm does not require a first guess. The tests reported in this chapter have motions in the x-direction with a phase of -1.3 radians relative to a sine where the x-axis is parallel to the bottom of the image. Motions in the y-direction have a phase of -1.4 radians relative to a sine. These phases were chosen to be consistent with the gradient-based method tests. Tests not reported in this chapter have shown a small dependence on the phases of the motions.

All noise simulations are identical to those of Chapter 5. Bias refers to the estimated value subtracted from the actual value.

6.5 Performance with No Duty Cycle

We consider motion estimates of a spot moving sinusoidally. The errors in Hanning-based estimates of amplitude for motions along one axis are shown in Figure 6-1 for 32x32 and 64x64 pixel regions. For the spot, the errors in amplitude estimation are approximately linear with amplitude. Doubling the dimensions of the region size shrinks the error by more than a factor of two. The pair-wise data showed that the Hanning estimator consistently returned estimates of the motion that scaled with the actual motion. The error in the phase estimator (not shown) was approximately 3×10^{-4} radians for all amplitudes for both region sizes.

We simulated motions of the image of the rod and spot shown in Figure 6-2. The simulations were identical to those of the bead. For both 32 by 32 pixel regions and 64 by 64 pixel regions, amplitude estimator errors were larger than 1 pixel (not shown). For many amplitudes, phase errors were larger than one radian (not shown).

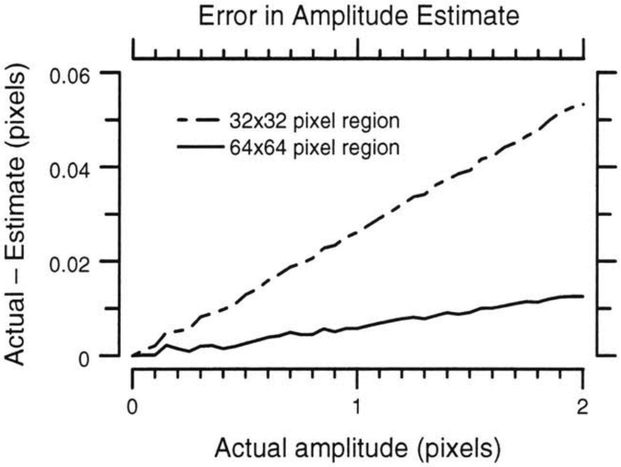

Figure 6-1: Error in the amplitude estimate of the spot along one axis using the Hanning-based estimator. Each trial consisted of eight simulated images of a moving spot acquired at evenly space phases of its sinusoidal motion. For each amplitude, the center of the spot was chosen randomly, using a uniform distribution in a square bounded by the center of four pixels in a 2x2 box.

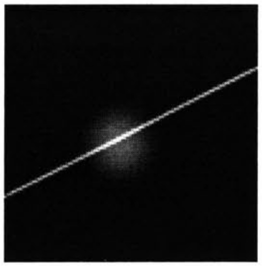

Figure 6-2: Image of a rod and a spot. The spot is an radially symmetric Hanning window of radius 20 pixels. The rod has a cross section of a Hanning window of length 6 pixels.

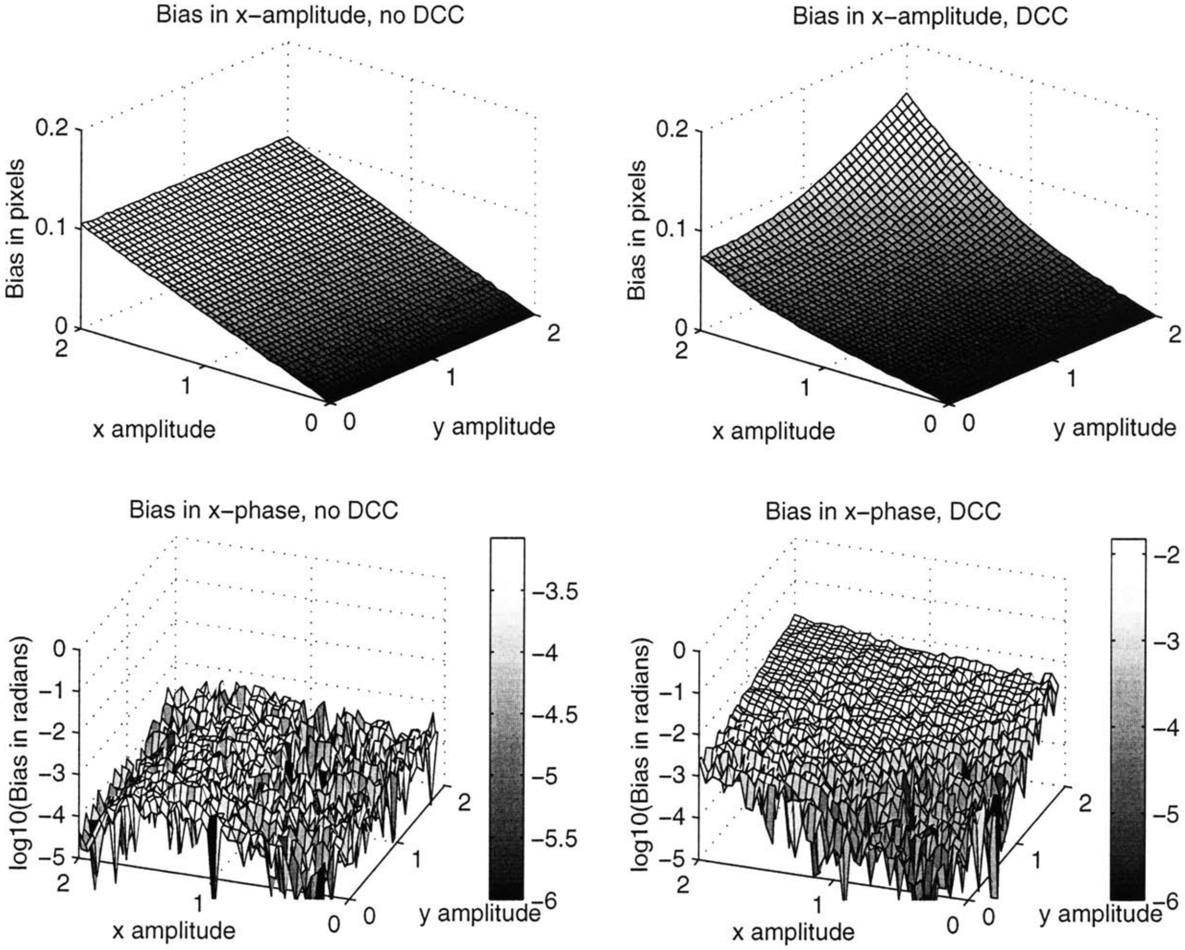

Figure 6-3: Bias in the estimate of amplitude and phase of the spot using a Hanning-based estimator with and without duty cycle compensation. For each test, eight evenly spaced images of the spot moving sinusoidally in time were simulated with $\frac{1}{8}$ duty cycle. The analysis region is 32x32 pixels. For each simulation, the center of the spot was chosen randomly, using a uniform distribution in a square bounded by the center of four pixels in a 2x2 box.

6.6 Performance with One Eighth Duty Cycle

We consider simulated images of the spot moving sinusoidally, acquired with $\frac{1}{8}$ duty cycle. In Figure 6-3 we examine the errors in estimates of amplitude and phase with and without duty cycle compensation (DCC) for a 32x32 pixel region surrounding the bead. The largest decrease in the amplitude estimator error due to DCC is 20%. However, for motions with amplitudes near 2 pixels in both directions, the amplitude estimator error increases. DCC significantly increases the error in the phase estimate for large motions.

Repeating the identical experiment with a 64x64 pixel region produces almost identical phase errors to those of the 32x32 pixel region. The errors in the amplitude estimate are similar to the 32x32 pixel analysis region, but scaled so that the maximum error for the non-DCC estimator is 0.06 pixels rather than 0.1 pixels. As it does for the smaller of the two region sizes, DCC at most decreases the error by 20%, but increases the error for amplitudes near 2 pixels in both directions.

We find that duty-cycle compensation has similar effects for images of the fatigue test device and the SAR image. DCC creates up to a 20% decrease in error in the amplitude estimate, but can increase the error in the amplitude estimate for large motions. For large motions, DCC increases the error in the phase estimate. We present no additional results using duty cycle compensation because of its marginal benefits.

6.7 Performance in the Presence of Noise

Figure 6-4 and Figure 6-5 show the errors and standard deviations of the Hanning window based estimators of amplitude and phase for the image of the fatigue test device in the presence of noise. Errors in amplitude vary linearly for both the 32x32 region size and the 64x64 region size. The errors using the larger region are approximately half the size of the errors using the smaller region size. Standard deviations for both cases are roughly 0.002 pixels. Phase errors range from less than 10^{-4} radians to 10^{-2} radians.

Simulations of the SAR image and the image of the bead yielded similar results to the fatigue test device. Phase errors ranged from less than 10^{-4} radians to 10^{-2} radians in both cases. For both region sizes of the SAR image, amplitude errors along an axis varied linearly with amplitude along that axis. For the bead, amplitude errors along an axis varied linirealy with amplitudes along that axis for the 32x32 pixel region only. We found that amplitude errors using the 64x64 pixel region surrounding the bead decreased by a factor of 2 over the 32x32 pixel region for most amplitude pairs, and not at all for others. Table 6.1 compares the maximum amplitude estimate errors and average standard deviations for all three images. In all cases the maximum error either decreased or remained the same with increasing analysis region size.

Image	32x32 pixel region		64x64 pixel region	
	Error	Stan. Dev.	Error	Stan. Dev.
Fatigue device	0.12	0.003	0.08	0.002
SAR Image	0.06	0.001	0.05	0.0003
Bead	0.11	0.01	0.11	0.01

Table 6.1: Maximum error and average standard deviation of the Hanning-based amplitude estimator. All numbers are in pixels. The simulations are described in Figure 6-4.

6.8 Discussion

In all simulations, in the absence or presence of noise, we found that increasing the region size decreased the error in the Hanning-based estimator. We believe this effect is due to the band

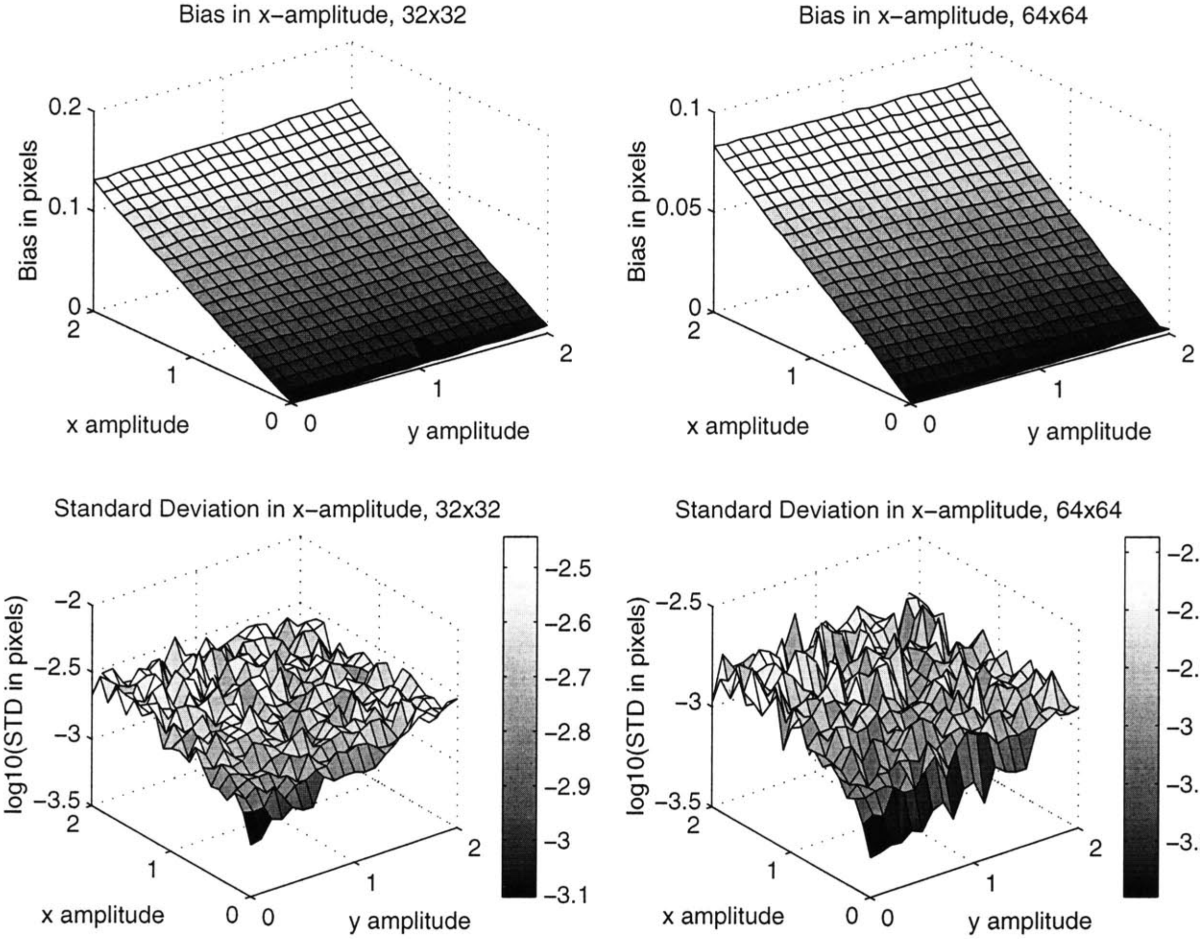

Figure 6-4: Bias and standard deviation in the estimate of amplitude of the fatigue device using a Hanning-based amplitude estimator for 32x32 and 64x64 pixel regions. Each trial consisted of 8 images of the device acquired with evenly spaced phases with duty cycle $\frac{1}{8}$ and -50 dB of shot noise and -50 dB of fixed-pattern added.

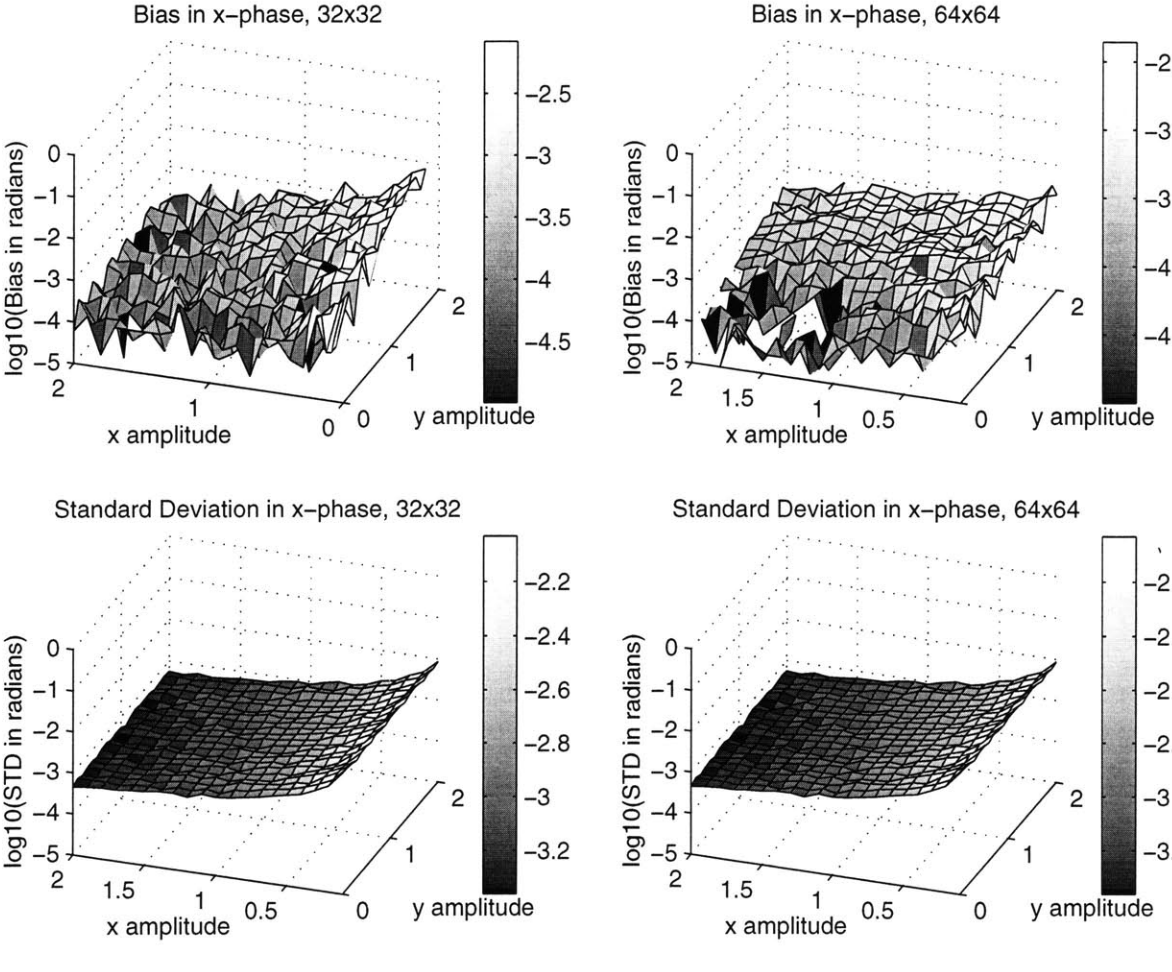

Figure 6-5: Bias and standard deviation in the estimate of phase for the fatigue test device using a Hanning-based amplitude estimator for 32x32 and 64x64 pixel regions. The simulation is described in Figure 6-4.

approximation discussed in Section 6.3.2. As the region size increases, the bands decrease in size and the spectrum in the band becomes more uniform so that the error decreases.

In all cases, the Hanning-based estimator underestimated the motion. By multiplying an image by a pattern, Hanning-based estimators effectively add fixed-pattern noise to the sample. For spectral estimation, window functions tradeoff precision and accuracy [27]. Window functions make the same tradeoff for motion estimation. By adding a pattern to the sample, we reduce variability in the results but lose accuracy.

It is not surprising that duty cycle compensation did not have as large a beneficial effect for the Hanning-based estimator as it did for the gradient-based methods in Chapter 5. The Fourier-based correlator is inherently biased so that at best DCC could remove the additional bias due to duty cycle effects. That DCC creates larger errors for large enough motions is also not surprising. Large motions produce temporal harmonics that are aliased into the first three harmonics of the motion as described in Chapter 5.

The simulation results of the image of the rod and spot illustrates that the Fourier-based estimators can make large errors depending on image content. The image was chosen to accentuate effects due to edge mismatches discussed in Section 6.3.3. The edge mismatches cause high frequencies which, though the sidelobes of the window function, cause bias in the motion estimate.

For 32x32 pixel region sizes, the errors in the Hanning window based amplitude estimator are roughly a factor of 2 larger than the errors of the LBC amplitude estimator examined in Chapter 5. For larger region sizes, it is possible that the Hanning-based estimator would outperform LBC. However, we did not investigate this possibility.

6.9 Conclusion

We developed a sinusoidal amplitude and phase estimator using Fourier-based correlation methods. We showed that using a Hanning windowed discrete Fourier transform on 32x32 pixel region yields amplitude errors less than roughly 0.1 pixels and phase estimates with better than 0.01 radian accuracy. However, by examining the image of the rod and the spot, we showed the possibilities for large error depending on image content.

The errors made by the Fourier-based estimator we developed are larger than the errors made by the linear bias corrected gradient-based algorithm developed by Davis and Freeman [7]. Because there exists a algorithm that makes smaller errors and because of the possibility for particularly large errors depending on image content, we choose not to examine the algorithm we developed further.

Chapter 7

Measurements of Sinusoidal Motions with Nanometer Accuracy using Multi-Image Gradient Methods and a Light Microscope,

7.1 Abstract

We investigate the use of multi-image gradient-based methods in conjunction with light microscopy to estimate the amplitude and phase of sinusoidal motion. We compare the errors made estimating amplitude using the multi-image gradient-based methods to those made by a linear bias corrected image registration algorithm developed by Davis and Freeman [8]. The multi-image based estimators achieve nanometer precision and accuracy estimates of amplitudes for motions smaller than 500 nanometers; ten times the maximum amplitude for which LBC achieves the same accuracy and precision. The multi-image gradient-based methods are shown to give results consistent to within a nanometer using different analysis regions of the same images. Computer simulations are shown to well approximate algorithm performance.

7.2 Introduction

Multi-image based gradient methods were developed as a technique to create algorithms whose performance is not limited by bias inherent to the algorithm. We used the technique to develop estimators of amplitude and phase of sinusoidal motion (Chapter 5). Simulations with noise conditions representing those found in optical microscopy show that errors below 0.001 pixels in amplitude are achievable for motions smaller than 1.2 pixels. For an optical microscope using a 20x objective and a CCD camera with 10 μm pixels, those simulations predict roughly 2 nanometer precision and accuracy.

Image resolution using light microscopy is limited to roughly the wavelength of light, typically 500 - 700 nm [16]. Any structure in an imaged target smaller than the resolution of the imaging system is blurred. However, because gradient-based algorithm rely on moving image contrasts, it is possible to detect motions of blurred images to a fraction of the blurring distance. Limitations on motion estimation are instead determined by the contrast of the target, the resolution and precision of the optics and camera, by sample vibrations and other noise sources.

In this paper, we compare the performance of two types of gradient-based algorithms. The first algorithm is a sinusoidal estimator based on linear bias corrected (LBC) two-image motion estimates [8]. The second is a class of algorithms based on multi-image filters. Simulations have shown that the class of multi-image gradient-based methods have significantly smaller inherent bias than the LBC algorithm.

We tested the accuracy and precision of the two types of gradient-based algorithms, comparing estimated motions to those given by a laser-vibrometer. We compared the experimental results to predictions based on computer simulations of noise processes. We examined the consistency of the two methods on different analysis regions of the same images. We also examined consistency between the different multi-image algorithms in the class.

7.3 Methods

Images of test structures were obtained using a combination of light microscopy and stroboscopic illumination. Images are corrected for the varying gains and offsets in the camera using a two-point correction technique. Motions are estimated by analyzing the corrected images with motion estimation algorithms. Results are compared to laser doppler measurements of motion and simulations of the algorithm.

7.3.1 Test Structures

To test our motion detection system, we used two test specimens. The first was a fatigue test structure designed by Exponent Failure Analysis Associates, Inc. This imaging target was chosen to represent our interests in the motions of microelectromechanical systems (MEMS). We mounted the chip containing the test structures on a piezo stack (AE0203D08, NEC, Japan) to move the chip perpendicular to the optical axis. The second test structure was a 1 μm polystyrene microsphere (Polysciences, Warrington, PA) which we refer to as a bead. The bead was chosen as a small signal to noise target, to test the limitations of our technique. The bead was attached to a 0.17 mm glass cover slip which was attached to a piezo stack (AE0203D08, NEC, Japan) so that the motions of the bead were perpendicular to the optical axis. In both cases, the piezo and CCD camera were aligned so that more than 99% of the motion was along one axis of the camera.

We drove both piezos with a 99.681 Hz sinusoidal stimulus voltage. The fatigue test device has a resonance at 20 kHz [13], and we measured the out of plane resonance of the glass cover slip to be 313 hertz. By driving both piezos at a frequency well below the resonance of each system, we expect the majority of the motions of each specimens were in-plane motions due to the piezo stimulus.

7.3.2 Video Microscopy

A light microscope (Zeiss, Axioplan, Thorwood, NY) imaged the bead using transmitted light, a long working distance condenser with 0.6 numerical aperture (0.6 NA) and a 20x, 0.4 NA objective with a 1x optivar. This configuration reflects our interests in biological applications. The fatigue test device was imaged using epi-illumination with the same light collection system. Images were recorded using a 1024x1024 pixel 12-bit CCD camera with 12 μm pixel spacing (DALSA CA-D7, Waterloo, Ontario). We measured the magnification of our imaging system to be 568.8 nanometers per pixel. The video microscope rested on a vibration-isolation table to attenuate floor vibrations.

7.3.3 Stroboscopic Illumination

The maximum frame rate of a scientific grade CCD camera is typically on the order of tens of hertz. We use stroboscopic illumination to image faster motions. A computer controlled current source pulses a green LED, (Nichia NSPG500S, Tokushima 774, Japan) so that a test sample is illuminated during a chosen phase of its motion. Light from the LED is collected using a halogen lamp housing with the LED positioned in place of the halogen lamp. The plastic lens that had covered the LED's die was milled away and the flat surface polished so that illumination was Köhler [20].

7.3.4 Image Analysis

Images are corrected for pixel-to-pixel sensitivity and offset variations using a linear fit to the response of each pixel. We use a two-point correction technique as illustrated in [8]. We collect a "dark" image to correct for varying offsets in the camera. After a microscope is set to image a specimen, between 50 and 128 images were collected with no sample illumination. We average these images to obtain the dark image. To compensate for the varying sensitivities of individual pixels, we attempt to image a uniformly bright region. To obtain "bright" images for the bead, we focused on the bead and then moved the glass cover slip to the side so that the light path from the condenser to the imaging lens passed through air only. To obtain bright images for the fatigue test device, we replaced the structure with a reflecting silicon wafer. For both cases, between 50 and 128 images were averaged to obtain a bright image.

For each measured image E, we create a corrected image G given by

$$G[i,j] = \frac{E[i,j] - D[i,j]}{B[i,j] - D[i,j]}$$

where $E[i,j]$ represents the gray value generated by the $(i,j)^{\text{th}}$ pixel in the measured image, and $B[i,j]$ and $D[i,j]$ represent corresponding gray values in the bright and dark images, respectively.

7.3.5 Motion Detection

We tested the pairwise linear bias compensated (LBC) image registration algorithm that was developed by Davis and Freeman [7] and then developed into a sinusoidal estimator [8]. We also tested four multi-image gradient-based algorithms listed in table 7.1. For each multi-image algorithm, there are two different methods for estimating sinusoidal amplitude and phase: the linear least square (LLS) fit technique and the discrete Fourier transform (DFT) method. Both methods are described Chapter 5. Because the results of the eight different calculations were found to be similar, we report results using the LLS method with algorithm 14 in all sections except Section 7.6, where we compare the performance of the different methods.

Name	x-Derivative Filter Support	y-Derivative Filter Support	t-Derivative Filter Support
algorithm 14	11x11x8	11x11x8	11x11x8
algorithm 16	19x19x8	19x19x8	19x19x8
algorithm 15	20x4x8	4x20x8	4x4x8
algorithm 17	36x4x8	4x36x8	4x4x8

Table 7.1: Filter support of derivative calculations for the multi-image algorithms. The x- and y- axes are aligned respectively along the bottom and side of each image. AxBxC refers to the dimensions, in pixels, of the box shaped filter support region, respectively along the x,y and t directions. The filters include duty cycle compensation and a 4-point low-pass filter as described in Chapter 5. For each algorithm, we use the LLS and DFT methods to estimate amplitude and phase as described in the text. Thus, there are a total of eight multi-image motion estimators.

In each test, the piezo was stimulated sinusoidally with constant amplitude and eight images evenly spaced in phase were collected. The LED was controlled so that the duty cycle was $\frac{1}{8}$ period. The tests were repeated 100 times to estimate the mean and standard deviation of each method.

Measurements of amplitude and phase were also made using a laser doppler interferometer (OFV511 with controller OFV3001, Polytec, Waldbronn, Germany). The laser was focused on a piece of reflecting tape attached to the end of the piezo. Data were sent to a spectrum analyzer (3562A, Hewlett Packard) to measure velocity amplitude and phase. Estimates of displacement amplitude were obtained by dividing the velocity amplitude by the frequency of the motion and 2π. Laser doppler measurements were performed both before and after the video data collection. We report the numbers before each experiment.

For each sample we examined a 32x32 pixel analysis region. The multi-image gradient-based algorithms use points outside the analysis regions in the support of their estimates; LBC does not. In order to compare the consistency of the algorithms on different contrast functions, we examined an additional 32x32 pixel region in the image of the fatigue test device. We used a second analysis region on the comb drive of the device as shown in Figure 7-1. We consider this second region only in Section 7.4.2.

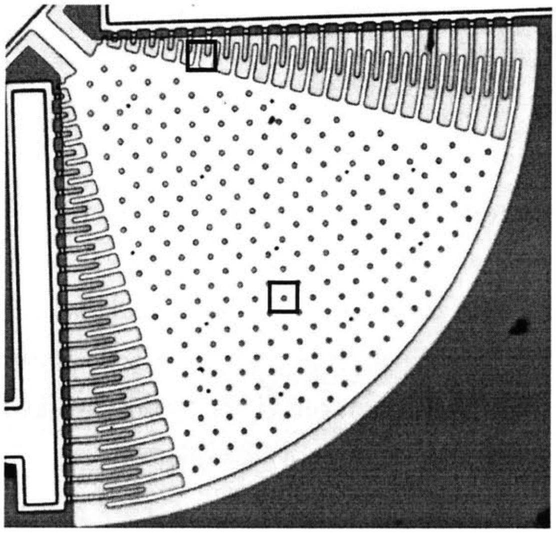

Figure 7-1: Image of the fatigue test device on the left and the bead on the right. For both images, motion was parallel to the bottom of the image. The two boxes on the fatigue test device indicate analysis regions. The analysis region on the combs of the fatigue test device was examined only in Section 7.4.2 for a consistency check.

All estimates are reported as amplitudes and phase along one axis only.

7.3.6 Simulations

To test our understanding of the performance-limiting mechanisms, we compared experimental results to simulations of theoretical models. The simulations included fixed-pattern noise, shot noise, quantization errors, and blurring.

The bead was found to be well simulated by a Hanning window given by

$$G[i,j] = \begin{cases} M_0(1 - \frac{1+\cos 2\pi r/6}{7}) & \text{for } |r| < 3, \\ M_0 & \text{otherwise} \end{cases} \tag{7.1}$$

where r represents the distance to the center of the spot and M_0 was chosen to adjust noise levels.

Motions of the fatigue test device were modeled using an image of the device in conjunction with a discrete Fourier transform, multiplication by a pure phase and an inverse transform.

For our camera, shot noise was measured to be −51 dB. Measurements of fixed-pattern noise depended on the size of the region used. For regions slightly larger than our analysis region, fixed-pattern noise was found on average to be −44 dB before two-point correction and −50 dB after two-point correction. We simulated identical fixed-noise patterns on ten sets of eight images. We simulated ten different randomly-generated fixed-noise patterns for a total of 100 simulations. Note that the simulations are inherently flawed in that they are based on many different fixed-pattern noises, instead of the single fixed-noise pattern in our camera.

The finite duty cycle of the strobe causes blurring of the moving objects in acquired images. Blur was simulated by averaging 100 images evenly spaced in time during the image acquisition period.

7.3.7 Calibration Errors

The laser doppler instrument is guaranteed to be calibrated only to within 1 percent. An additional multiplicative factor in the calibration of the laser doppler interferometer is possible due to the misalignment of the laser with the motion of the piezo. We aligned the laser by maximizing its reported signal. We found several local maxima in signal amplitude each within 1% of the others. Due to alignment and calibration errors, it is possible that the motions reported using the laser doppler interferometer and the motions estimated using the image processing algorithms will disagree by as much as several percent. We are interested in errors of roughly 1 nanometer. We expect differences between the laser doppler and video measurements due to calibration errors to be less than a nanometer for amplitudes smaller than roughly 50 nanometers.

7.4 Results

7.4.1 Amplitude Accuracy and Precision

In Figure 7-2 we plot the error in the estimate of amplitude by algorithm 14 and LBC for images of the fatigue test structure. The shaded regions represent ± one standard deviation of the simulations of the error. For motions smaller that about 50 nanometers, the multi-image algorithm errors are between 1 and 2 nanometers. For the same range, LBC errors are in the range of 10 nanometers. For small motions the simulations and experiment overlap; however, for larger motions, the simulations and errors diverge. We find that increasing the laser doppler estimate by 1% makes the simulated and experimental values agree better. In Figure 7-3 we multiply the laser doppler measurement by 1.01. Assuming this calibration correction of the laser doppler measurement, algorithm 14 achieves nanometer resolution for all measurements.

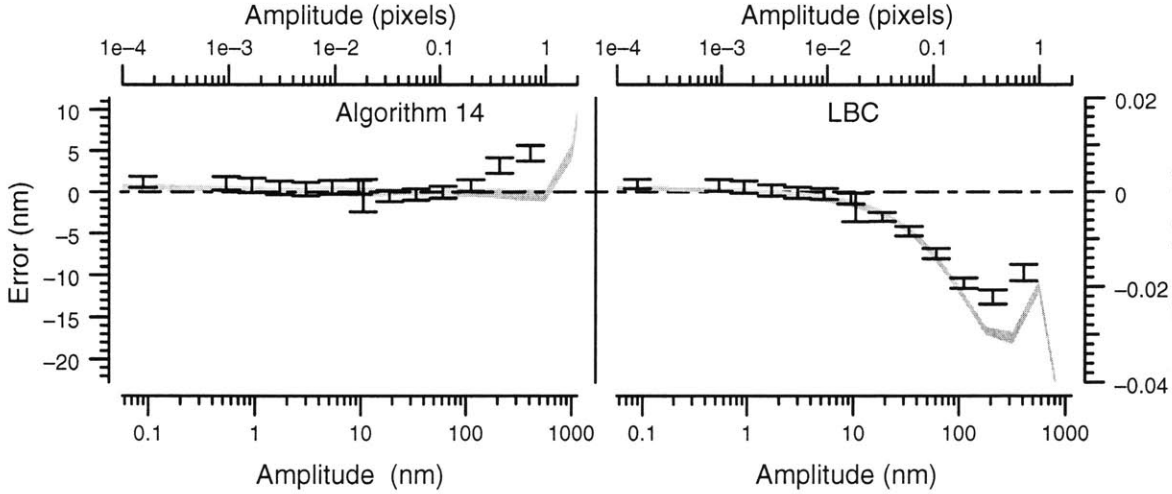

Figure 7-2: Error in the estimate of amplitude by algorithm 14 and LBC as a function of the laser doppler estimate of amplitude for the central analysis region of the fatigue test device. Error refers to the algorithm estimate subtracted by the laser doppler estimate. The error bars are the square root of the sums of the variances of the algorithm and the laser doppler amplitude. The large error bar at 10 nanometers is due to a single outlier in the data. The grey regions are the ± one standard deviation regions predicted by simulations. A reference line at zero error is shown for clarity.

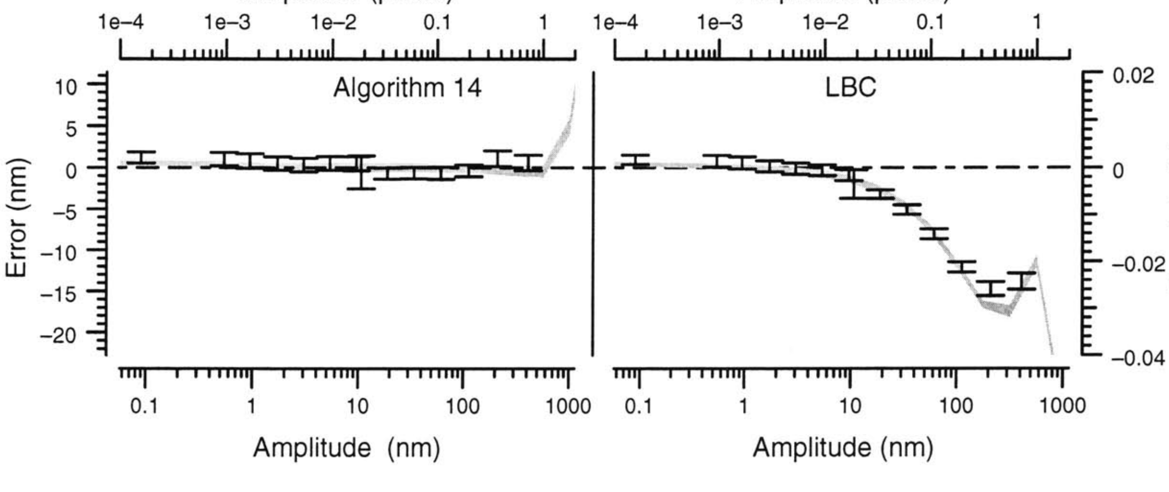

Figure 7-3: Error in the estimate of amplitude by algorithm 14 and LBC as a function of the laser doppler estimate of amplitude for an image of the fatigue test device. The plot is identical to Figure 7-2 except we have multiplied the laser doppler measurements by 1.01.

In Figure 7-4 we plot the error for algorithm 14 and LBC for the bead as we did in Figure 7-2. The experiment and simulations overlap. LBC loses nanometer accuracy roughly above 10 nanometers and then diverges quickly reaching a maximum error of 20 nanometers. Algorithm 14 loses nanometers accuracy near 50 nanometers. However, algorithm 14's error slowly diverges reaching a maximum of only 5 nanometers.

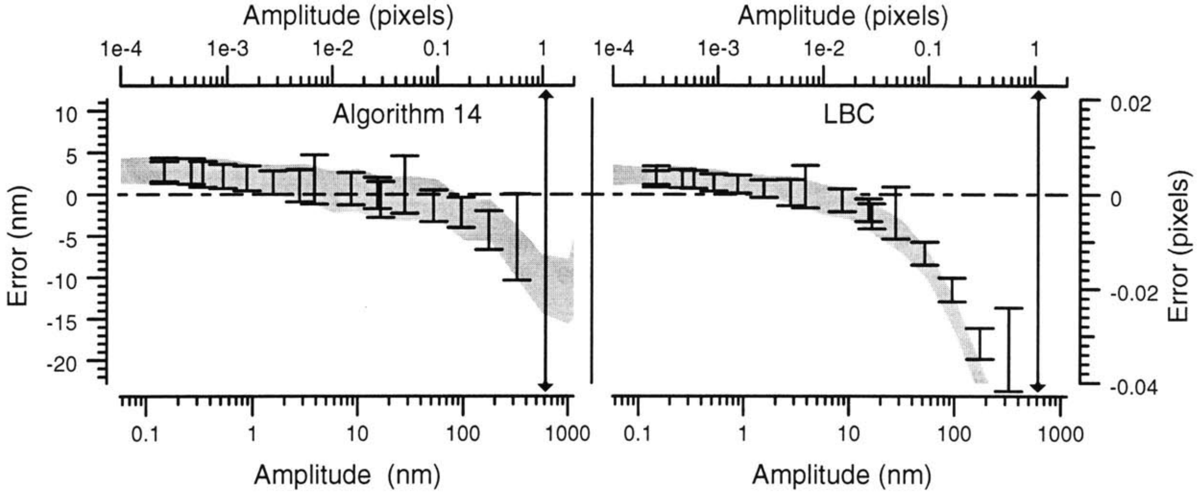

Figure 7-4: The error in the estimate of amplitude by algorithm 14 and LBC as a function of the laser doppler amplitude for an image of a single bead. Other aspects of this figure are as in Figure 7-2.

Figure 7-5 shows the precisions of algorithm 14 and LBC are between 1 and 3 nanometers for images of the bead for all measurements. Similar results were found for the fatigue test device. The simulated precisions roughly overlay with the experimental precisions up to 80 nanometers. For larger amplitudes, the simulations appear to diverge from the data exponentially. The divergence is actually roughly linear, appearing exponential because of the semi-logarithmic plot.

7.4.2 Consistency

An important measure of accuracy is the difference in motion estimates using different analysis regions. One would like an algorithm that gives the same answer independent of the content of the analysis region. In Figure 7-6 we show the difference between estimates of motion for two different regions of the fatigue test device. LBC gives answers that differ in mean by more than 5 nanometers. Conversely, estimates from algorithm 14 for the two different regions differed in mean by less than 2 nanometers. The other seven multi-image algorithms performed as well or better than algorithm 14 .

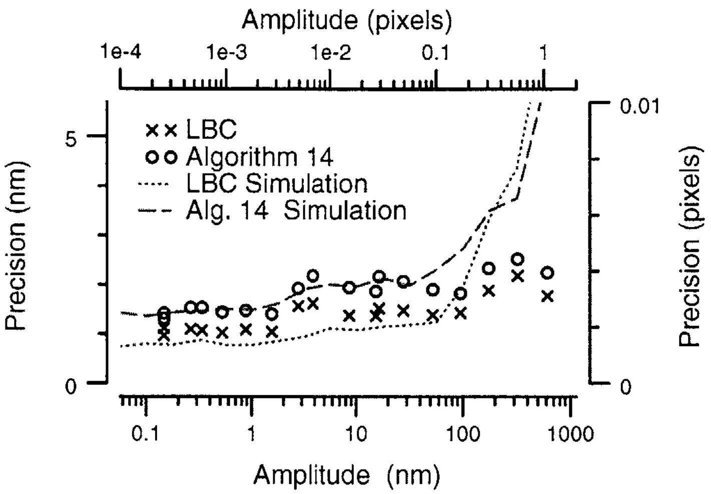

Figure 7-5: The precision (standard deviation) of algorithm 14 and LBC as a function of the laser doppler estimate of amplitude. Experimental values are shown as "x" and "o". Simulated values are shown as lines.

7.5 Phase

Except for the largest motions, the standard deviation of phase estimates for LBC and algorithm 14 dominate the accuracy of both algorithms. Figure 7-7 shows the precision of the phase estimates as a function of amplitude. The precisions are roughly the same and agree with experimental predictions. A similar plot for the fatigue test device shows the same shape, with both algorithms yielding roughly a factor of two decrease in standard deviation at all amplitudes.

7.6 Consistency Among Multi-Image Based Methods

In Figure 7-8, we plot the difference in the amplitude estimate between algorithm 14 and the seven other algorithms at a function of the amplitude estimate of algorithm 14 for the central analysis region in the fatigue test device. For the 9,800 points in the figure (14 test voltages, 100 tests, and 7 algorithms) the maximum difference between all eight algorithms for any motion estimate is 0.6 nanometers. The difference between each DFT and LLS estimate using the same filters (not shown) is almost a factor of ten smaller than the difference between motion estimates using difference filters.

7.7 Discussion

We have shown that for the fatigue test structure, the multi-image gradient-based estimators and LBC yield estimates of motion with nanometer precision and accuracy. Figure 7-3 shows that algorithm 14 retains nanometer accuracy up to 500 nanometers while LBC retains the same accuracy up to only 10 nanometers. LBC is limited by bias inherent to the algorithm that exists even in the

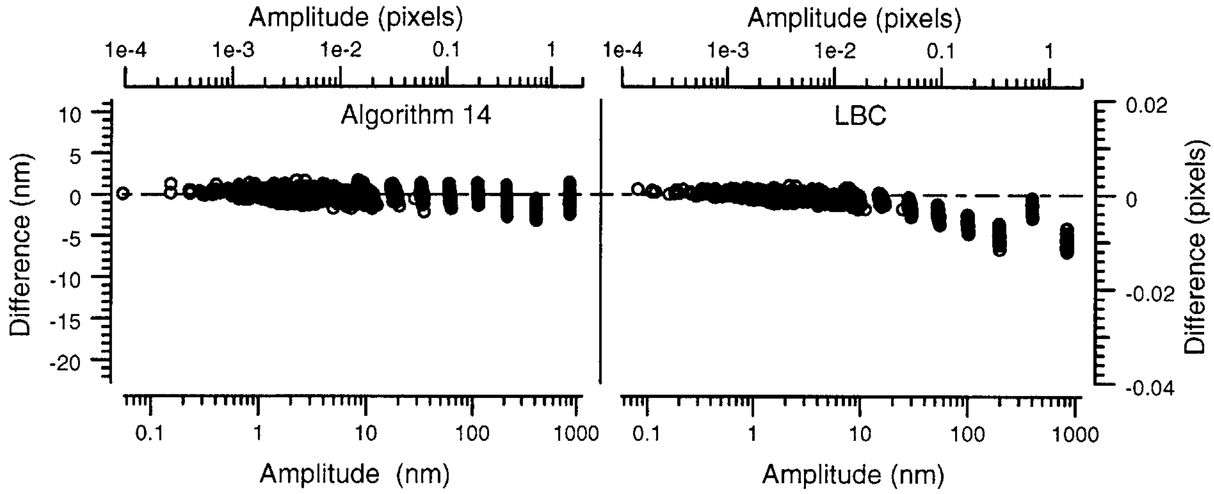

Figure 7-6: Difference in amplitude estimates between two regions of the fatigue test device as a function of the estimate in one of the regions. The first region is in the interior of the device; the second region focuses on the comb drive of the device as shown in Figure 7-1

absence of noise [7]. Conversely, the multi-image gradient-based methods were designed to have inherent bias below 0.001 pixels up to amplitudes slightly larger than a pixel. The inherent bias in LBC accounts for the difference in the performance of the two algorithms.

The bead has a radius of roughly 3 pixels so that the great majority of the analysis region around the bead contains no signal. Even under these conditions, algorithm 14 is able to estimate motions with errors less 5 nanometers. Conversely, LBC's error approaches 20 nanometers for the same data set.

The precision of the two algorithms is comparable. Figure 7-5 shows that LBC has a roughly 20% smaller standard deviation at all amplitudes. That LBC has the smaller amplitude is not surprising. For estimation techniques, there is typically a tradeoff between accuracy and precision. We feel that the accuracy gain in the multi-image algorithms over LBC is well worth the 20% increase in standard deviation over LBC.

Our simulations accurately predicted precision and bias with the exception of the precision of the amplitude estimator as shown in Figure 7-5. For large motions, the simulations of amplitude precision predict linearly larger variability than is found in the data set. Because our simulations included ten different fixed noise-patterns that were not identical to the fixed-noise pattern of our camera, we expect our simulations to yield somewhat different results from the actual experiment. Therefore, we do not find the linear divergence surprising.

That the simulations predict the majority of the errors and precision of the algorithms suggests that the noise factors simulated, shot noise and fixed-pattern noise, are in fact the dominate noises in the algorithms. If, for example, table vibrations played a large role, we would have expected significantly larger standard deviations than what we found.

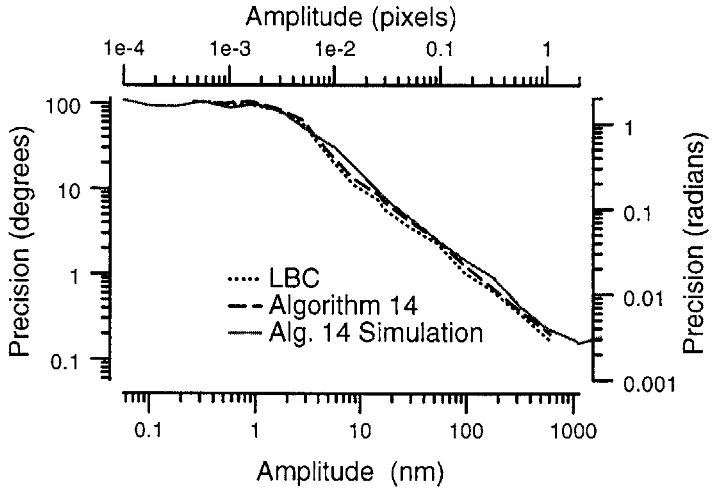

Figure 7-7: Phase Precision (one standard deviation) for each algorithm for the image of the bead as a function of the laser doppler measurement of amplitude. The simulated results are for algorithm 14; LBC's simulated precision is very similar. No efforts were made to handle phase wrapping errors.

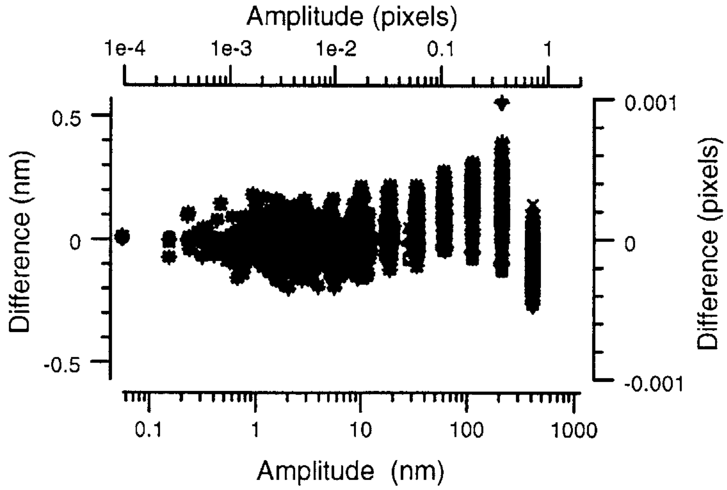

Figure 7-8: Amplitude dependence of the difference between amplitude estimates using algorithm 14 and the other seven multi-image algorithms. The analysis region was the central region of the fatigue test device as indicated in Figure 7-1

The consistency of the gradient-based algorithms over two regions with very different spectral content is remarkable (see Figure 7-6). It is equally remarkable that all eight algorithms give estimates of motion within 0.6 nanometers of each other. The algorithms were designed to be consistent for regions for images with most of their energy below spatial frequency 2. The algorithms estimate derivatives using very different filters with lengths that differ by more than a factor of 3. That the algorithms perform so closely supports our assumptions that derivatives can be well estimated using filter-based methods.

It is not surprising that the standard deviation of the phase estimator increases inversely with amplitude. Phase is calculated using the ratio of the amplitudes of the in-phase and out-of-phase components of the motion. As the two components shrink, their precision remains roughly constant (see Figure 7-5) so that the fractional uncertainty in the ratio should increase inversely proportional to amplitude magnitude.

7.8 Conclusion

We have shown that multi-image gradient-based estimators of amplitude can achieve nanometer accuracy and precision for motions smaller than 500 nanometers. LBC gives errors a factor of 10 larger over the same range so that the new methods provide significant improvements over LBC. We have shown that the multi-image algorithms are consistent to within a nanometer across different analysis regions of the same images. Finally, we have shown that our simulations using fixed-pattern noise and shot noise capture the majority of our experimental findings.

Chapter 8

Summary

Our goal was to create an estimator of temporal sinusoidal motion that was not limited by errors inherent in it. We investigated widely-used methods to estimate motion and chose two approaches to achieve our goal. The first approach was based on gradient methods. The second approach was based on Fourier transforms.

8.1 Multi-Image Gradient-Based Algorithms

We created a new class of multi-image gradient-based algorithms. We showed that these algorithms estimate velocity, not displacement. We also showed that the inherent bias of these algorithms can be made arbitrarily small for small enough motions. However, for large enough motions, the algorithms make large errors. These errors are due to temporal changes in brightness that are not sampled sufficiently to meet the Nyquist criteria.

Using the multi-image gradient-based algorithms, we created estimators of the amplitude and phase of temporal sinusoidal motion. We simulated the performance of these estimators for images of a spot, a microelectromechanical device and a synthetic aperture radar image of the Galapagos Islands. In the presence of noise typical of our image acquisition systems, −50 dB of shot noise and −50 dB of fixed pattern noise, the estimators achieved errors below 0.001 pixels and 0.001 radians for amplitudes smaller than 1.2 pixels. However, for motions larger than roughly 2 pixels, the amplitude errors were larger than 0.1 pixels.

We tested the accuracy of our methods by comparing motions estimated using the algorithms to those measured by a laser doppler interferometer. We examined the motions of a fatigue test structure and a bead attached to a piezo. For the fatigue test structure, the algorithms achieved nanometer accuracy and precision for motions below 500 nanometers. The simulations well predicted the performance of the algorithms; the estimator achieved near 0.001 pixel accuracy and precision. We also showed that the algorithms are consistent to within a nanometer across regions of the same moving object with very different brightness distributions. For a moving bead, an object chosen for its particularly small signal to noise ratio, we showed that the algorithms achieved five nanometer accuracy and precision.

We compared the accuracy of the multi-image gradient based algorithms to those of the linear bias corrected algorithm (LBC) created by Davis and Freeman [7]. We showed that these methods also achieved nanometer precision and accuracy, but only for motions below roughly 10 nanometers. Thus, the range of high accuracy estimation for the multi-images algorithms is a factor of 10 bigger than the range for LBC. However, unlike our estimator, LBC works well for motions larger than 2 pixels.

8.2 Fourier Transform Based Methods

We developed a Fourier transform based motion estimator. However, for the 32x32 pixel regions that interest us, this estimator was found to be limited by bias inherent in the estimator. We examined an image of a rod and a spot, chosen to accentuate edge effects, and found that the estimator could make errors greater than 1 pixel in amplitude and 1 radian in phase. Simulations in the presence of noise for more representative images of our applications showed the Fourier transform-based methods achieved 0.1 pixel accuracy and 0.01 radian accuracy. These inaccuracies are greater than those of already existing algorithms, such as LBC.

8.3 Conclusion

The multi-image gradient-based algorithms achieved our goal of making a sinusoidal estimator that, for small enough motions, is not limited by bias inherent to the method.

We believe these algorithms will be of great practical importance to the field of microelectromechanical systems (MEMS). High accuracy measurement tools are vital for the test and validation of MEMS. This work also has applications for measuring bio-mechanical motions, such as sound-induced motions of inner ear structures. More accurate measurement tools will help elucidate the mechanisms that underlie our remarkably sensitive hearing.

Our methods take advantage of the increasing availability of CCD cameras and computational power to obtain quantitative motion measurements from an optical microscope. These measurements transforms optical microscopes from qualitative imagers to quantitative motion measurement systems.

Appendix A

Second Order Gradient Methods with Perfect Derivative Estimation

Recall (3.7)

$$0 = \nabla \mathrm{E} \cdot \widehat{\boldsymbol{\delta}}_r + \mathrm{E}_t \delta_t + \frac{1}{2}\widehat{\boldsymbol{\delta}}_r^T \mathrm{E}_{Hr} \widehat{\boldsymbol{\delta}}_r + \frac{1}{2}\mathrm{E}_{tt}\delta_t^2 + \delta_t \nabla \mathrm{E} \cdot \widehat{\boldsymbol{\delta}}_r + \dots \tag{A.1}$$

where $\mathrm{E}_{Hr} = \nabla_r(\nabla_r \mathrm{E})$ is the spatial Hessian of the continuous intensity function E, E_t is the temporal derivative of E and ∇ is the spatial gradient operator. Let $\mathrm{E}(\boldsymbol{r}, t) = \mathrm{E}(\boldsymbol{r} - \boldsymbol{d}(t), 0)$. In this case, we can write the equations in simpler form by separating out the spatial and temporal derivatives. We find

$$\mathrm{E}_t = -\nabla \mathrm{E} \cdot \boldsymbol{d}'(t), \tag{A.2}$$

$$\nabla \mathrm{E}_t = -\mathrm{E}_{Hr} \cdot \boldsymbol{d}'(t), \text{ and} \tag{A.3}$$

$$\mathrm{E}_{tt} = -\nabla \mathrm{E} \cdot \boldsymbol{d}''(t) + \boldsymbol{d}'^T \mathrm{E}_{Hr} \boldsymbol{d}'(t). \tag{A.4}$$

Using these three equations, we rewrite (A.1) as

$$\begin{aligned} 0 = {} & \nabla \mathrm{E} \cdot \left[\widehat{\boldsymbol{\delta}}_r - \boldsymbol{d}'(t)\delta_t - \frac{1}{2}\boldsymbol{d}''(t)\right] \\ & + \frac{1}{2}\left[\widehat{\boldsymbol{\delta}}_r^T \mathrm{E}_{Hr} \widehat{\boldsymbol{\delta}}_r + 2\delta_t \widehat{\boldsymbol{\delta}}_r^T \mathrm{E}_{Hr} \boldsymbol{d}'(t) + \delta_t^2 \boldsymbol{d}'^T(t) \mathrm{E}_{Hr} \boldsymbol{d}'(t)\right] + \dots . \end{aligned} \tag{A.5}$$

We would like a solution of the form $\widehat{\boldsymbol{\delta}}_r = \boldsymbol{d}'(t)\delta_t + \frac{1}{2}\boldsymbol{d}''(t)\delta_t^2$. Substituting this trial solution into (A.5), we find

$$0 = \frac{1}{8}\delta_t^4 \boldsymbol{d}''^T(t)\mathrm{E}_{Hr}\boldsymbol{d}''(t) + \ldots \tag{A.6}$$

So the solution we desire is correct up to order δ_t^2. The assumption of gradient-based methods is small motions, which we quantify by requiring the first term in the Taylor expansion of the motion to be much larger than the second,

$$\boldsymbol{d}'(t)\delta_t >> \boldsymbol{d}''(t)\delta_t^2 \tag{A.7}$$

Thus the error in (A.6) is small compared to other terms.

Bibliography

[1] Arking A and Lo RC. A Fourier approach to cloud motion estimation. *Journal of Applied Meteorology*, 17(6):734–744, 1978.

[2] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images—a review. *Proc. IEEE*, 76(8):917–934, 1988.

[3] J. L. Barron, D. J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *Journal of Computer vision*, 12:43–77, 1994.

[4] Philip R. Bevington and D. Keith Robinson. *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill, Boston, MA, 1992.

[5] J.A. Bloom and T. R. Reed. An uncertainty analysis of some real functions for image processing applications. In *Proceedings. International Conference on Image Processing*, volume 3, pages 670–673, Los Alamitos, CA, 1997. IEEE Comput. Soc.

[6] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* John Wiley & Sons, New York, NY, 1991.

[7] C. Q. Davis and D. M. Freeman. Statistics of subpixel registration algorithms based on spatio-temporal gradients or block matching. *Optical Engineering*, April 1998.

[8] C. Q. Davis and D. M. Freeman. Using a light microscope to measure motions with nanometer accuracy. *Optical Engineering*, April 1998.

[9] C. Q. Davis, Z. Z. Karu, and D. M. Freeman. Equivalence of subpixel motion estimators based on optical flow and block matching. In *IEEE International Symposium for Computer Vision*, pages 7–12, Coral Gables, Florida, November 1995.

[10] Charles Quentin Davis. *Measuring Nanometer, Three-Dimensional Motions with Light Microscopy.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1997.

[11] S. C. Douglas. A frequency-domain subpixel position estimation algorithm for overlay measurement. *SPIE*, 1926:402–411, 1993.

[12] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5(1):77–104, 1990.

[13] D. M. Freeman, A. J. Aranyosi, M. J. Gordon, and S. S. Hong. Multidimensional motion analysis of MEMS using computer microvision. In *Solid-State Sensor and Actuator Workshop*, pages 150–155. Transducer Research Foundation, Inc., June 1998.

[14] A. Gatherer and T. H.-Y. Meng. Robust subpixel alignment in lithography. *J. Vac Sci. Technology B.*, 10(6):2662–2666, 1992.

[15] B Girod. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications*, 41(4):604–612, 1993.

[16] J. W. Goodman. *Introduction to Fourier Optics.* McGraw-Hill Book Company, New York, 1968.

[17] D.J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1(4):279–302, 1987.

[18] B. K. P. Horn and E.J. Weldon, Jr. Direct methods for recovering motion. *Internatl. J. of Computer Vision*, 2:51–76, 1988.

[19] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[20] Shinya Inoué. *Video Microscopy.* Plenum Press, New York, NY, 1986.

[21] James R. Janesick, Tom Elliot, Stewart Collins, Morley M. Blouke, and Jack Freeman. Scientific charge-coupled devices. *Optical Engineering*, 26(8):692–714, 1987.

[22] C. H. Knapp and G. C. Carter. The generalized correlation method for estimation of time delay. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.

[23] U-V Koc and KJR Liu. DCT-based motion estimation. *IEEE Transactions of Image Processing*, 7(7):948–965, 1998.

[24] U-V Koc and KJR Liu. Interpolation-free subpixel motion estimation techniques in DCT domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(4):460–87, 1998.

[25] C Kuglin and D Hines. The phase correlation image alignment method. In *Proceedings of the 1975 International Conference on Cybernetics and Society*, pages 163–165, San Francisco, CA, September 1975. IEEE.

[26] E. Lantz. Subpixel signal centering and shift measurement using a recursive spectral phase algorithm. *Signal Processing*, 17:365–372, 1989.

[27] J.S. Lim. *Two Dimensional Signal and Image Processing.* Prentice Hall, New Jersey, 1990.

[28] H. Liu, T. Hong, H. Martin, T. Camus, and R. Chellappa. Accuracy vs. efficiency trade-offs in optical flow algorithms. *Computer Vision and Image Understanding*, 72:271–286, 1988.

[29] H. Liu, T-H Hong, M. Herman, and R. Chellappa. A generalized motion model for estimating optical flow using 3-d hermite polynomials. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 1, pages 361–6, San Francisco, CA, 1994. IEEE Comput. Soc. Press.

[30] H. Liu, T-H Hong, M. Herman, and R. Chellappa. A general motion model and spatio-temporal filters for computing optical flow. *International Journal of Computer Vision*, 22(2):141–172, 1997.

[31] H.H. Nagel. On the estimation of optical flow: relations between different approaches and some new results. *Artificial Intelligence*, 33(3):229–324, 1987.

[32] H.H. Nagel. Optical flow estimation and the interaction between measurement errors at adjacent pixel positions. *International Journal of Computer Vision*, 15(3):271–88, 1995.

[33] L. Ng and V. Solo. Errors-in-variables modeling in optical flow problems. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 2773–2776, New York, NY, 1998. IEEE.

[34] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, 1989.

[35] P. Sprent. A generalized least-squares approach to linear functional relationships. *Journal of Royal Statistical Society Series B*, 28:2, 1966.

[36] Q. Tian and M.N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233, 1986.

[37] Lloyd N. Trefethen and David Bau Trefethen. *Numerical Linear Algebra.* Society for Industrial and Applied Mathematics, Philadelphia, 1997.

[38] Wu-Ki Tung. *Group Theory in Physics.* World Scientific, Philadelphia, PA, 1985.

[39] Paul A Viola. *Alignment by Maximization of Mutual Information.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.

[40] Willsky. Class notes from Stochastic Processes, Detection and Estimation (6.432). Soon to be published.