29 March 2001

Multi-Image Gradient-Based Algorithms for Motion Estimation

Samson J. Timoner and Dennis M. Freeman

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

1 Abstract

Image registration algorithms based on gradient methods provide quantitative motion measurements from sequences of video images. Although such measurements can be degraded by image noise, larger degradations typically result from systematic bias in the algorithms that is present even if the images are noise-free. To improve the accuracy of motion measurements, we have developed a new class of multi-image algorithms based on multi-dimensional digital filters. The new algorithms provide better estimates of spatial and temporal gradients and also compensate for motion blur caused by the non-zero acquisition time of the imager. We have optimized filters for measuring arbitrary motions, and we illustrate results when those filters are used to estimate constant velocity movements. We also show results for filters that are optimized for harmonic analysis of periodic motions. Using these algorithms, systematic bias in the amplitude of sinusoidal motion is less than 0.001 pixels for motions smaller than one pixel in amplitude. This represents a hundred-fold decrease in bias compared to existing methods.

Keywords: optical flow, registration, subpixel, gradient, tracking, spatio-temporal filters

2 Introduction

Recent advances in CCD imaging technology have made high quality video imagers widely available. In combination with a computer, video imagers allow not only qualitative observations of motion but also quantitative measurements. Our goal is to take advantage of this combination to enable high resolution motion analysis of biological and artificial micromachines that are otherwise difficult targets of study [1; 2; 3; 4; 5; 6].

Our approach is to estimate motions from sequences of images obtained using stroboscopic illumination [3]. While motions can be estimated by processing sequences of images pairwise, processing larger groups of images can be advantageous. For example, combining multiple images into higher level constructions, such as "mosaics," can lead to an efficient and complete representations that can be useful for video compression, video indexing, search, and manipulation [7]. Furthermore, processing sequences naturally leads to iterative schemes that can dramatically improve accuracy and reduce computational costs [8]. One mechanism that leads to increased accuracy is that increasing the number of frames in the estimation procedure tends to increase the number of constraints [9; 10]. Our goal is to take advantage of multiple images to develop algorithms to estimate motions with subpixel accuracy.

Our algorithms are based on spatio-temporal gradients [11; 12; 13; 14]. Unlike edge detection methods [15] or point-correspondence methods [16; 17], algorithms based on spatio-temporal gradients can be used to register arbitrary images: no prior knowledge of the target is necessary. Furthermore, these algorithms take advantage of information from all parts of the image (rather than just at edges or other features), which is particularly important for the registration of noisy images.

Many groups have examined multi-image spatio-temporal filters for motion estimation. Several groups estimate velocity using tuned Gabor filters [18; 19; 20; 21]. Other groups examined multi-image gradient methods [22; 23; 24]. These authors generally address the complications of detecting arbitrary motion fields, occlusion, and robustness to small local gradients ("the aperture problem").

We have focused on a different issue: how each step in a motion detection algorithm affects the ultimate resolution of the resulting estimates. Specifically, we focus on determining the relationship between gradient errors and motion estimation errors.

Many physical factors contribute to errors in motion estimates based on video images, including shot noise due to the quantum nature of light, Johnson noise in electrical amplifiers, fixed-pattern noise due to pixel-to-pixel changes in imager sensitivity, and mechanical stability of the measurement apparatus [3; 25]. Algorithmic factors are also important. In fact, errors due to intrinsic bias in algorithms based on first difference approximations to gradients [11] are typically larger than those due to physical factors for modern scientific-grade cameras [26].

We view motion estimation as a signal processing problem. To remove the algorithmic limitations, we developed a new class of multi-image algorithms based on multidimensional digital filters. Because filter errors decrease roughly exponentially with filter support [27], a small increase in filter support can lead to significantly more accurate gradient estimates, and therefore motion estimates.

The filters can be optimized for specific applications. We demonstrate general motion detection algorithms. Also, because the measurement of sinusoidal motions is so important in measuring the modes of mechanical systems, we create algorithms specialized for harmonic analysis of periodic motions. Simulations of the new methods show that the systematic bias in the amplitude of sinusoidal motion is less than 0.001 pixels for motions smaller than one pixel in amplitude. This represents a hundred-fold decrease in bias compared to first-difference methods. In fact, systematic errors due to bias in the new algorithms are reduced to the point that they are tiny compared to random errors due to noise, even for images from low-noise, scientific grade, CCD imagers.

3 Motion Estimation Algorithms

Translations of a scene cause brightness changes that are related by the “constant brightness assumption” [11; 12] which can be expressed as

$$E_x(x,y,t)u_x(t) + E_y(x,y,t)u_y(t) + E_t(x,y,t) = 0 \tag{1}$$

where $E_x(x,y,t)$, $E_y(x,y,t)$ and $E_z(x,y,t)$ represent gradients of the image brightness $E(x,y,t)$ in the x, y, and t directions and $u_x(t)$ and $u_y(t)$ represent the velocities of the scene at time t in the x and y directions, respectively. Let $G[i,j,k]$ represent samples of $E(x,y,t)$ that would be obtained with a video camera with $x = i\Delta$ and $y = j\Delta$, where Δ is the pixel spacing of the camera, and $t = kT$, where T is the temporal sampling period. Equation 1 can be approximated as

$$G_x[i,j,k]v_x[k] + G_y[i,j,k]v_y[k] + G_t[i,j,k] = 0 \tag{2}$$

where $G_x[i,j,k]$, $G_y[i,j,k]$, and $G_t[i,j,k]$ are functions of $G[i,j,k]$ that represent discrete approximations to $E_x(x,y,t)$, $E_y(x,y,t)$, and $E_t(x,y,t)$, and $v_x[k]$ and $v_y[k]$ represent discrete approximations to $u_x(t)$ and $u_y(t)$. If at some time the image gradients in Equation 2 are known for $0 \leq i < I$, and $0 \leq j < J$, then the equation represents a set of $I \times J$ equations linking the two unknown velocities: v_x and v_y. We solve this overconstrained set of equations using the method of least squares, yielding

$$\begin{bmatrix} \hat{v}_x \\ \hat{v}_y \end{bmatrix} = - \begin{bmatrix} \sum G_x G_x & \sum G_x G_y \\ \sum G_x G_y & \sum G_y G_y \end{bmatrix}^{-1} \begin{bmatrix} \sum G_x G_t \\ \sum G_y G_t \end{bmatrix} \tag{3}$$

where the sums are over i and j, $\hat{v}_x$ and $\hat{v}_y$ are estimates of the velocities v_x and v_y, and all values are at time k.

3.1 Gradient Estimation

We estimate the gradients $G_x[i,j,k]$, $G_y[i,j,k]$, and $G_t[i,j,k]$ by convolution of the discrete images $G[i,j,k]$ with linear filters. For example, convolution with

$$h_1[i] = \delta[i+1] - \delta[i] \tag{4}$$

where $\delta[0] = 1$ and $\delta[i] = 0$ for $i \neq 0$, provides a first difference approximation to the gradient of a one-dimensional function. This first difference filter best approximates spatial gradients between pixels. To obtain a gradient estimate at a pixel, one can average two first difference estimates to obtain

$$h_2[i] = (\delta[i+1] - \delta[i-1])/2. \tag{5}$$

To satisfy Equation 2, all three gradients must be estimated at one location in space and time. (Examples of problems with gradients that are not co-located can be found elsewhere [27; 28; 24].) If the filter in Equation 5 is used to estimate all three gradients, then the gradients are co-located at a pixel. However if the first difference filter is used, the gradients do not satisfy the co-location condition. To co-locate gradients in multiple dimensions, we introduce interpolation filters. For example

$$h_3[i] = (\delta[i+1] + \delta[i])/2 \tag{6}$$

provides a first order approximation of the value of a function at points between pixels. In general, we let

$$\begin{aligned} G_x[i,j,k] &= (g_x[i] \;\; h_y[j] \;\; h_t[k]) * G[i,j,k] \\ G_y[i,j,k] &= (h_x[i] \;\; g_y[j] \;\; h_t[k]) * G[i,j,k] \\ G_t[i,j,k] &= (h_x[i] \;\; h_y[j] \;\; g_t[k]) * G[i,j,k] \end{aligned}$$

where $g_x[i]$, $g_y[j]$, and $g_t[k]$ represent gradient approximations, $h_x[i]$, $h_y[j]$, and $h_t[k]$ represent interpolators, and $*$ represents convolution.

3.2 Spatial Filters

Images from a video microscope are low-pass functions of spatial frequency because of the finite aperture of the optical train [29]. By controlling the numerical aperture of the objective and total magnification of the optical train, one

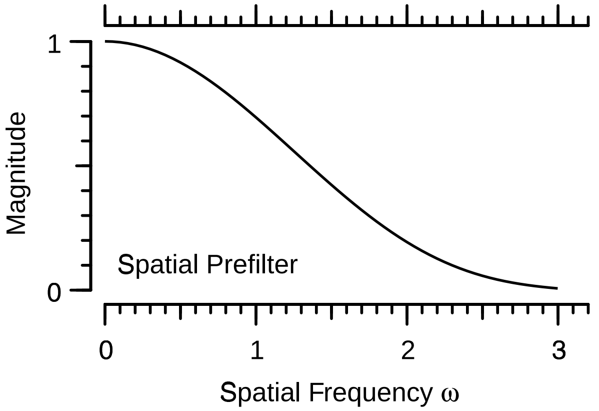

Figure 1. Spatial low-pass filter. The curve shows the magnitude of the frequency response of a symmetric, 4-point, low-pass filter used to attenuate high frequency image noise.

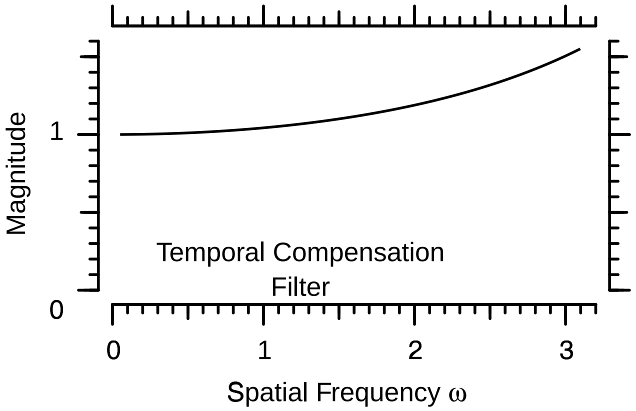

Figure 2. Temporal compensation filter. The curve shows the ideal magnitude of a filter that would compensate for non-zero acquisition time for an imager. The acquisition time is taken to be equal to the temporal sampling period.

can limit the maximum spatial frequency projected onto the camera, and thereby avoid "aliasing" [30]. An additional source of low-pass filtering results because each pixel in a CCD camera collects light from a portion of the image with an area on the order of the square of the interpixel distance (i.e., "fill factors" are typically close to 1). This non-zero sampling area tends to decrease high spatial frequencies [30]. Effects of this low-pass filtering could be compensated by using inverse filters. However, such compensation is not necessary for accurate motion estimation. Because the same linear filtering is applied to each image in a sequence, the spatial filtering is equivalent to changing the spatial frequency content of the target without changing its motion.

Unlike the image of the target, imaging noise due to the quantum nature of light and due to Johnson noise in the amplifiers tends to be broadband. For the majority of our microscope images, signal energy decreases with spatial frequency and is smaller than the noise energy for spatial frequencies above 2, where π is the Nyquist frequency. The difference in spatial frequency content between the signal and noise encourages two actions. First, we use a spatial low-pass filter (Figure 1) to attenuate high frequency noise (as in a Weiner filter). Second, we ignore high frequencies when designing the gradient estimation filters. This allows us to create more accurate filters in the spectrum of interest.

3.3 Temporal Filters

Images are obtained by integrating the light that falls on an imager during a non-zero acquisition time. The non-zero acquisition time smoothes temporal changes in brightness, blurs moving objects, and leads to errors in motion estimates [3]. Such errors can be reduced by incorporating knowledge of the acquisition process into the motion estimation algorithm. For example, if the imager collects light uniformly during the image acquisition time, then temporal changes in brightness will simply be low-pass filtered; the cutoff frequency of the filter decreases as the image acquisition time increases. Suppose that motions of a target cause changes in brightness with temporal radian frequency ω. If the image acquisition time is equal to the sampling period (so that the imager collects light striking its surface during the entire sampling period) then the non-zero acquisition time reduces the apparent magnitude of the brightness changes during the cycle by a factor of $\frac{\sin(\omega/2)}{\omega/2}$ [30]. Such changes in brightness can be compensated by inverse filtering with

$$H(e^{j\omega}) = \frac{\omega/2}{\sin\omega/2}$$

which is shown in Figure 2.

3.4 Filter Design

The Parks-McClellan algorithm [31; 30] creates filters that minimize a weighted error in the frequency domain with a weighting function that determines the relative importance of an error at a particular frequency. For spatial filters, we set the weighting function to be ω^{-1} for frequencies below 2 and 0 for frequencies above. An ideal interpolating filter has a magnitude of 1 at all frequencies ω; an ideal differentiator has a magnitude equal to ω [30]. Filters were calculated using the Matlab 5.1 "remez" function [32] and are listed in the appendix. Examples are shown in Figure 3.

Figure 3 illustrates the fundamental tradeoff between odd and even filters. Derivative filters of odd length best approximate the derivatives at points co-located with pixel centers. However, due to their symmetry, the magnitudes of such filters are 0 at $\omega = \pi$. Derivative filters of even length best approximate the derivatives at points midway between pixel centers. The magnitudes of such filters better approximate ideal derivative filters at high frequencies than do those with odd length. However, symmetry requires that the magnitudes of interpolation filters of even length are 0 at $\omega = \pi$. Conversely, the best odd interpolation filter is simply $h[n] = \delta[n]$, which is exactly the ideal interpolation filter. Thus, even length filters better approximate derivative filters than odd length filters of comparable length. However, odd length filters (which are always of length 1) better approximate interpolation filters than do even length filters, which are always longer.

It is possible to change the filter accuracy tradeoff by considering the pre-filter. The pre-filter can be implemented by convolving it with both the spatial derivative and spatial interpolation filters. In the frequency domain, this is equivalent to multiplying the spectrums of the gradient and interpolation filters by the spectrum of the pre-filter. The resulting spectrums need not resemble those of ideal interpolation or differentiation filters. Only the ratio of the spectrums must remain near the ideal ratio, $\frac{1}{\omega}$.

One can create much more accurate filters by designing interpolation and gradient filters together, rather than separately. A small error in the gradient filter spectrum can be corrected by a corresponding identical error in the interpolation filter spectrum. We design spatial filters by minimizing a weighted error of the ratio of the spectrums of the interpolation and gradient filters. We use a squared error with a weighting function that is the same as that used with the Park-McClellan algorithm except at low frequencies. The new weighting function was set to $16/\pi$ for all frequencies less than $\pi/16$. This change produced significantly improved the performance of the resulting filters. We initialize our minimization using the Park-McClellan filters convolved with the pre-filter. We then search for a local minimum in the weighted error of the ratio using the simplex method [33]. The error in ratio of a few example filters are shown in Figure 4; the filters are given in the appendix. The shapes of the spectrums of the individual filters are unremarkable and are not shown.

The same method of design does not work for temporal filters. Changing the pre-filter in space changes the image of the target without changing its apparent motion. Changing the pre-filter in time changes the apparent motion of the target.

3.5 Small Motions

Motions of a target cause changes in pixel brightnesses that depend on both the temporal properties of the motion and the spatial properties of the scene. For example, large changes in pixel brightness can result from either large displacements of a low contrast target or from small displacements of a high contrast target. Temporal changes in pixel brightness are particularly simple for motions that are small compared to the distances over which the spatial gradients in brightness change significantly. For such motions, temporal changes

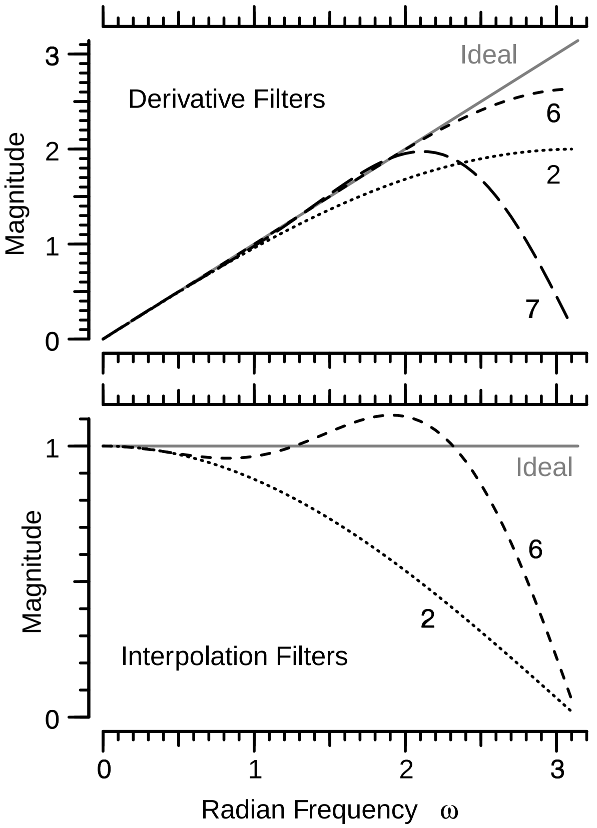

Figure 3. Derivative and interpolation filters. The gray lines show the magnitudes of the frequency responses of an ideal derivative (top) and interpolation (bottom) filter. The dashed lines show the magnitudes of frequency responses of the 6- and 7-point filters designed with the Parks-McClellan algorithm. The dotted lines show the magnitude responses of the first order filters given by Equations 4 and 6.

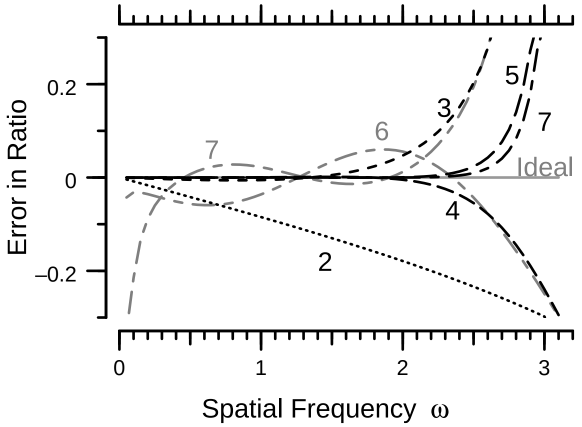

Figure 4. The error in ratio of spectrums of filters designed by considering only the ratio of the spectrums. The black lines show filters that were created by minimizing the weighted, squared error of the ratio of the spectrums. The gray lines show filters created using the Parks-McClellan algorithm. The 2-point filters (Equations 4 and 6) are shown for reference.

in pixel brightness are proportional to temporal changes in target displacement (proportionality constant is the spatial gradient in target brightness). In contrast, large motions generate brightness changes at frequencies that are related to the spatial frequency content of the target. Consider a target that consists of a single spatial frequency and that is moving with constant velocity v. Brightness changes as $\sin(\omega(x - vt) + \phi))$. Thus, a spatial frequency ω leads to a temporal frequency ωv.

Temporal filters are designed using the Parks-McClellan algorithm, with a weighting function of $\frac{1}{\omega}$ from $0 \ldots 2$ and 0 elsewhere. By excluding high temporal frequencies, we are able to make significantly more accurate filters in the remainder of the spectrum. We choose the ω^{-1} weighting so that for small motion, and therefore small temporal frequencies, the filters are particularly accurate. The gradient and interpolation filters are made to include the pre-filter. The resulting filters are shown in Figure 5. Notice that the length 8 derivative filter is a significantly better approximation to the ideal than is the length 7 derivative filter.

3.6 Periodic Motions

For periodic motion, changes in brightness are periodic, consisting of a linear combination of a fundamental frequency and its harmonics. The temporal filters need to be accurate in only a small bandwidth around those frequencies. For the case of eight images per cycle, those frequencies are $\frac{\pi}{4}$, $\frac{\pi}{2}$, $\frac{3\pi}{4}$, π where π is the Nyquist frequency. We define 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. We use an ω^{-1} weighting as before. 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, as discussed in Section 3.4. The gradient and interpolation filters are made to include the pre-filter. An example 8-point filter is shown in Figure 5.

Since the motions are assumed to be periodic, we allow the temporal filters to wrap, using the first picture as if it followed the last and the last picture as if it preceded the first. Eight velocities are estimated at times midway between sampling times. The amplitude and phase of the first three and a half velocity harmonics are estimated using a discrete Fourier transform of the eight velocities. To determine the displacement amplitude and phase, we divide the velocity amplitude by the driving frequency and shift the phase by 90 degrees.

3.7 Volumes of Support

We test multi-image gradient-based algorithms with gradient filter designs of four different volumes of support: $3 \times 3 \times 8$, $5 \times 5 \times 8$, $7 \times 7 \times 8$, and $9 \times 9 \times 8$ where $x \times y \times t$ represents the number of pixels in the x, y, and t directions that are used to calculate each of the gradients in Equation 3. The $9 \times 9 \times 8$ algorithm uses spatial filters that are convolutions of 6-point Parks-McClellan filters with the spatial pre-filter. The remaining algorithms use spatial filters designed using the ratio minimization. All of these algorithms use the same 8-point temporal filter designed using the Parks-McClellan algorithm. In one case, we vary the temporal support and test algorithms with support $7 \times 7 \times 8$,$7 \times 7 \times 7$,$7 \times 7 \times 5$,$7 \times 7 \times 4$, and $7 \times 7 \times 2$. We compare the results for the multi-image gradient-based algorithms to results from the pairwise linear bias compensated (LBC) image registration algorithm for a sinusoidal estimator [26; 3] and to results for a first difference algorithm in which $g_x[i]$, $g_y[j]$, and $g_t[k]$ are given by Equation 4 and $h_x[i]$, $h_y[j]$, and $h_t[k]$ are given by Equation 6 [11; 12].

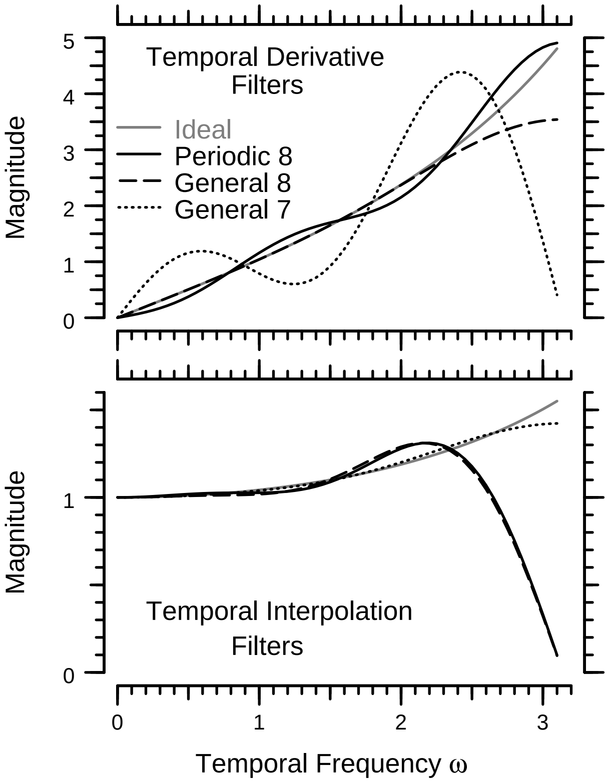

Figure 5. Temporal filters. The gray curves show the magnitudes of “ideal” temporal derivative (top) and interpolation (bottom) filters that represent the products of the ideal filters shown in Figure 3 with the temporal compensation filter shown in Figure 2. The continuous black curves show the magnitudes of 8-point discrete filters designed to optimize responses for the fundamental frequency and higher harmonics that result during periodic motion at $\omega = \frac{\pi}{4}$. The dashed and dotted black curves show the magnitudes of 8- and 7-point filters designed for arbitrary motions.

4 Methods

Motion estimation algorithms are tested by applying them to estimate motions of computer-generated images and computer-generated shifts of measured images. Computer-generated noise processes are applied to the images to simulate effects of imaging noise.

4.1 Images

We analyze motion estimates for three different test images (Figure 6). The first is a dark spot on a bright background. This image is defined mathematically by

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

where r is the distance from pixel i, j to the center of the spot and M_0 controls the background intensity. The small feature size provides little signal energy and the bright background produces large amounts of shot noise (described below). Thus, this image tests the algorithms under low signal-to-noise conditions. Furthermore, the mathematical representation allows its center to be placed at arbitrary locations in the image — on or between pixel centers. To avoid artifacts that could result if the spot were always aligned with the pixel grid, the center of the spot is chosen randomly (x and y coordinates uniformly distributed ± 1 pixel around the center of the image) for each test. The second image is a light micrograph of silicon micromachine designed by Exponent Failure Analysis Associates. This image represents our interests in metrology of microelectromechanical systems (MEMS). The third image is a synthetic aperture radar (SAR) image of the Galapagos Islands[1]. This image has significant energy in high spatial frequencies and tests the robustness of the algorithms to high spatial frequencies.

4.2 Translations

Translations of the simulated spot are accomplished by recalculating images using Equation 7. Translations of the fatigue test device and SAR images are accomplished by taking the discrete Fourier transform of the 512 by 512 pixel images, multiplying by space-shifting filters ($e^{j\Delta_x\omega_x + j\Delta_y\omega_y}$) and then inverse transforming. To avoid artifacts near the boundaries of the image, gradients are evaluated in only the center 32 by 32 pixel region (so that only the center 40 by 40 pixels are used for even the longest of the filters tested). The analysis regions are shown in Figure 6

To simulate finite image acquisition time, we average 100 images evenly spaced in time during the acquisition period. For the fatigue test structure and SAR images, we transform the images, multiply by the average of 100 different space-shifting filters, and then inverse transform.

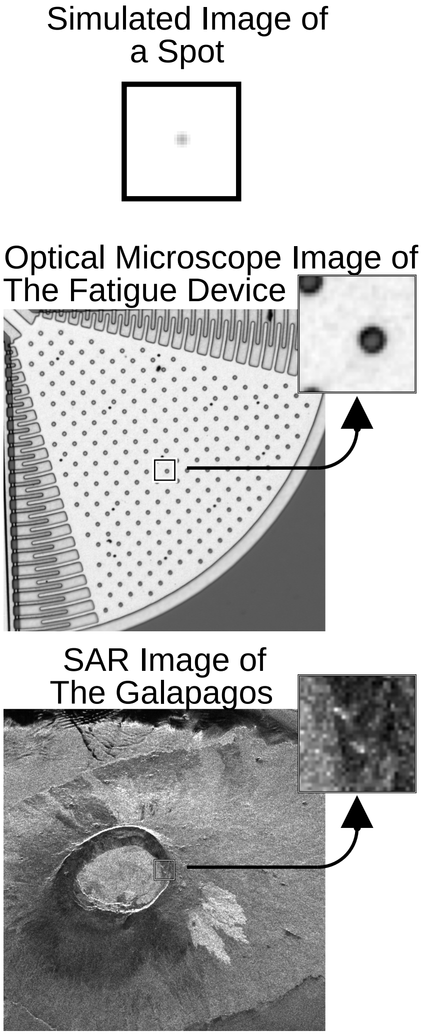

Figure 6. Test images. The top panel shows a simulated image of a spot (Equation 7). The middle panel shows an optical micrograph of a fatigue test structure. The bottom panel shows an SAR image of the Galapagos Islands. The inset in each panel shows a magnified view of the analysis region.

For each constant velocity test, motion is parallel to the x-axis only. For each sinusoidal test, displacements in the x-direction are given by $A_x \sin(\frac{\pi}{4}t - 1.3)$ and displacements in the y-direction are given by $A_y \sin(\frac{\pi}{4}t - 1.4)$, where A_x and A_y are the displacement amplitudes.

4.3 Simulated Noise

We simulate the effects of imaging noise, including fixed-pattern noise, shot noise and quantization errors. Motion is simulated by creating a sequence of eight high-precision images representing the average number of electrons collected at each pixel. Fixed-pattern noise is simulated by

[1] Image (P-43899) courtesy of the Jet Propulsion Laboratory, Pasadena, CA.

multiplying these images by a spatial array of pseudo-random gain factors taken from a Gaussian distribution (mean = 1, standard deviation = 0.00315) simulating fixed pattern noise with energy that is -50 dB below the average signal energy. The same spatial array is used for each of the eight images within one motion sequence. Shot noise is simulated using a pseudo-random Poisson number generator to convert the average number of electrons to an integer representing the number of electrons in one randomly chosen instance. The number of electrons is then divided by 32 (the number of electrons needed to increment the A/D output of a camera by one) and truncated (to model quantization errors).

All noise simulations are repeated using 10 different sequences of images for each of 10 different fixed-noise patterns. Different shot noise patterns are used for each image in each of the 100 sequences. The average brightness in the analysis region of each of the initial images is normalized so that the shot noise is -50 dB smaller than the signal energy.

5 Results

To investigate errors in motion estimates based on video images, we simulate the performance of four classes of algorithms: first differences [11; 12], LBC [26; 3], multi-image gradient-based methods specialized for periodic motion, and multi-image gradient-based methods generalized for arbitrary motion. Simulations in the absence of imaging noise reveal systematic biases that limit the ultimate resolution of the motion estimation algorithm. Studies with imaging noise illustrate the relative performance of each algorithm for particular imaging conditions as well as conditions when bias, noise, or both are important.

5.1 Bias in Estimates of Sinusoidal Motions

Four multi-image algorithms were designed to estimate sinusoidal motions of a target (Section 3.6). Filters for three of the algorithms ($3\times3\times8$, $5\times5\times8$, and $7\times7\times8$) were designed by minimizing errors in the ratio of the gradient and interpolation filters (Section 3.4). For comparison, filters for the remaining algorithm (9×9×8) were designed using the Parks-McClellan algorithm. Differences among these algorithms, the first difference algorithm, and LBC were assessed using noise-free images of the spot (Equation 7) undergoing sinusoidal motion. Results are shown in Figure 7.

For each algorithm, errors in estimating the magnitude of the motion tend to increase with the amplitude of the motion. However, trends for small and large motions differ. For small displacement amplitudes (less than 1 pixel), the bias is (1) a nearly linear function of displacement amplitude for each algorithm, (2) more than an order of magnitude smaller for multi-image algorithms than it is for the first-difference or LBC methods, and (3) more than two orders of magnitude smaller for the $5\times5\times8$ and $7\times7\times8$ algorithms than it is for the first-difference or LBC methods. For large displacement amplitudes (greater than 3 pixels), the magnitude bias is similar for the first-difference and multi-image methods and is significantly smaller for LBC.

Biases for the $5\times5\times8$ and $7\times7\times8$ algorithms are very similar. We find that the the performance of the two algorithms is significantly different only when processing images of high spatial-very content such as the Galapagos Islands. For the SAR image of the Galapagos Islands (not shown), the $7\times7\times8$ algorithm performs 10 times better than the 5×5×8 algorithm for small motions. The 9×9×8 algorithm, which uses spatial filters created with the Parks-McClellan algorithm rather than the ratio-minimization, performs as well as the $3\times3\times8$ algorithm and significantly worse than the $5\times5\times8$ and $7\times7\times8$ algorithms. Because of their relatively poor performance using small filter supports, multi-image gradient-based algorithms whose spatial filters are created with the Parks-McClellan algorithm are not explored in the remainder of this paper.

Sinusoidal motions of the spot image sequence were also estimated using multi-image algorithms with filters designed for arbitrary motions (Section 3.5). The resulting biases (not shown) were generally larger than those for algorithms that were specialized for periodic motions: amplitude biases were typically 10% greater and phase biases were approximately 3 times larger.

Noise-free images of the fatigue device undergoing sinusoidal displacements were simulated to assess errors in motion estimates using the multi-image algorithm with a 7×7×8 volume of support (Figure 8). Errors in estimating the amplitude of the x-component of displacement (A_x) depend on both A_x and A_y. Amplitude errors are less than 0.001 pixels for motions with total magnitudes $\sqrt{A_x^2 + A_y^2}$ less than 1.2 pixels. For larger motions, the amplitude errors gradually increase, becoming as large as 0.05 pixels for $A_x = A_y = 2$ pixels. For a given value of A_y, errors tend to increase with A_x. There is exceptions to this trend; for example, errors are unusually small for $A_x = 1$ pixel when $A_y = 0$. The atypically small error is the boundary where the amplitude estimates switch from being too large to being too small. Errors in estimating the phase of the x-component of displacement also depend on both A_x and A_y. Phase errors are particularly small for $A_y \approx 0.4$ pixel, marking the location where the error switches from negative to positive. Phase errors are smaller than 10^{-2} radians for all $A_x < 2$ and $A_y < 2$ pixels dropping to well below 10^{-4} radians for small motions. Similar trends are observed for analyses of the spot and SAR image sequences (not shown).

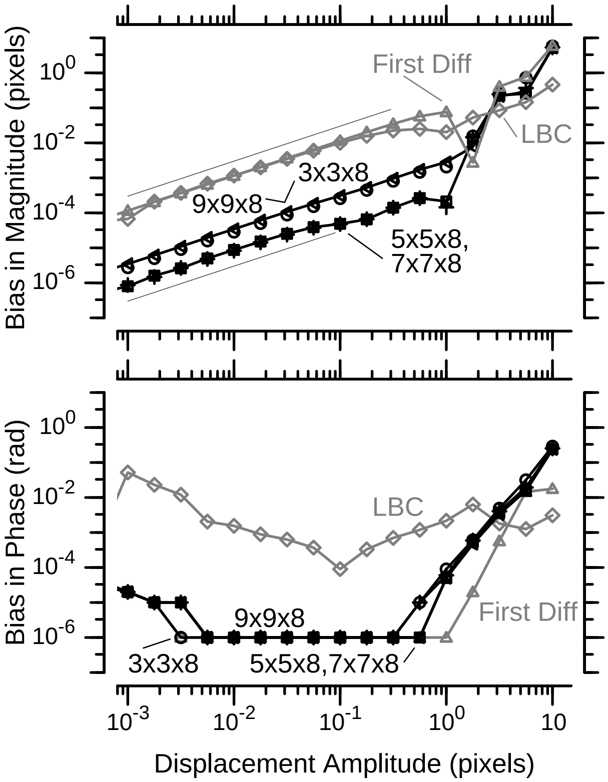

Figure 7. Amplitude dependence of bias in estimates of sinusoidal motion. Sinusoidal displacements of the spot (Equation 7) were analyzed for for 17 displacement amplitudes (Section 4.2, $10^{-3} < A_x < 10$, $A_y = 0$). For each amplitude, 8 images were generated using a sampling period equal to one-eighth of the period of the sinusoid and an image acquisition period equal to the sampling period. Symbols in the top panel represent the difference between the estimated amplitude of the displacement and the amplitude used to generate the images. Symbols in the bottom panel represent the difference between the estimated phase of the motion and the phase used to generate the images. Labels identify the algorithms: first differences, LBC, and periodic specific, multi-image methods with 4 volumes of support ($3 \times 3 \times 8$, $5 \times 5 \times 8$, $7 \times 7 \times 8$, and $9 \times 9 \times 8$). For reference, straight lines with unit slope are shown in the top plot. Phase biases smaller than 10^{-6} are plotted as 10^{-6} for clarity.

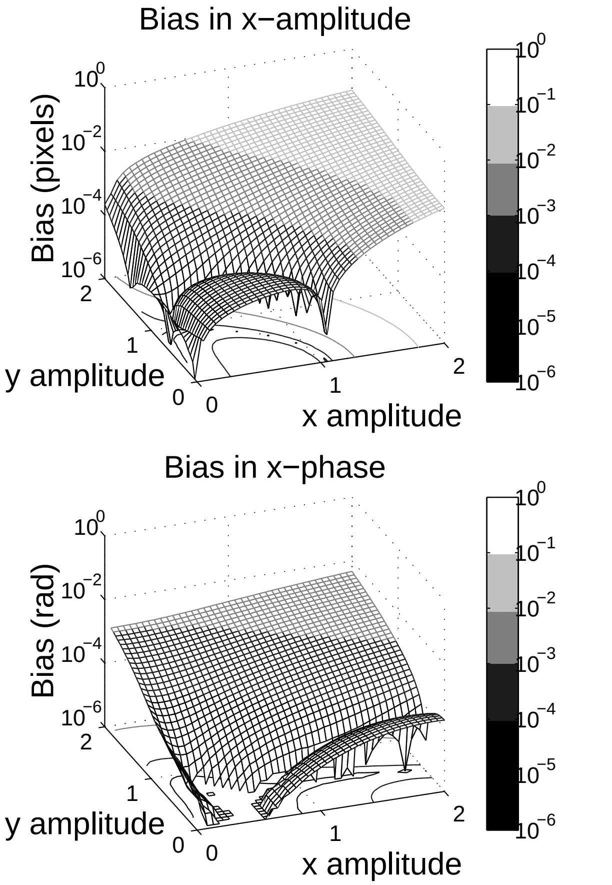

Figure 8. Amplitude dependence of sinusoidal motion estimates using the periodic specific 7×7×8 multi-image algorithm. Sinusoidal displacements of the fatigue device were analyzed for for 1682 pairs of displacement amplitudes A_x and A_y (Section 4.2). For each amplitude, 8 images were generated using a sampling period equal to one-eighth of the period of the sinusoid and an image acquisition period equal to the sampling period. The upper surface map illustrates the difference between the estimated amplitude of displacement in the x direction and A_x as a function of both A_x and A_y. The lower surface map illustrates the difference between the estimated phase of displacement in the x direction and the x phase used to generate the images. The contours show lines of constant bias (integral powers of 10). For clarity, the surface maps are shaded (right scale) to indicate bias.

5.2 Bias in Estimates of Uniform Motion

Eight multi-image algorithms were designed to estimate arbitrary motions of a target (Section 3.5). Four of the algorithms have the same temporal filters and different spatial support (3×3×8, 4×4×8, 5×5×8, and 7×7×8); the remaining filters have the same spatial filters and different temporal support (7×7×2, 7×7×4, 7×7×5, and 7×7×7). Differences among these algorithms, the first difference algorithm, and LBC were assessed using noise-free images of the spot, fatigue test device, and Galapagos Island undergoing uinform (constant velocity) motion. Results are shown in Figure 9.

Many of the trends for algorithms using 8-point temporal filters (Figure 9, left panels) are similar to those seen in Figure 7. For example, errors tend to increase with the velocity of the motion. Also, trends for small and large velocities differ. For small velocities (less than 0.5 pixel/frame), the bias is a nearly linear function of displacement amplitude for each algorithm and roughly an order of magnitude smaller for multi-image algorithms than it is for the first-difference or LBC methods. The biases of the 5×5×8 and 7×7×8 algorithms are at least two orders of magnitude smaller than the bias of LBC or the first difference algorithm. For the fatigue test structure and the Galapagos, bias is smaller for multi-image methods with larger volumes of support. However, for the spot image, the 3×3×8 algorithm outperforms both the 5×5×8 and $7 \times 7 \times 8$ algorithms. The performance of the $4 \times 4 \times 8$

algorithm (not shown) falls between the 5×5×8 and the 3×3×8 algorithms.

For the largest velocities (> 1 pixel/frame), the bias for LBC is much smaller than that for the multi-image methods or for the first difference algorithm.

For the spot and the Galapagos, the bias for the 7×7×8, 7×7×4, and 7×7×2 algorithms, are identical for asymptotically small motions. These algorithms have the same support in space but different support in time. For larger motions, the biases in the algorithms differ by up to a factor of 10. Note that the first difference algorithm ($2 \times 2 \times 2$) and the $7 \times 7 \times 2$ algorithm differ only in their spatial filters. The biases of the multi-image algorithms with odd support in time, the 7×7×7 and $7 \times 7 \times 5$ algorithms, is many orders of magnitude larger than the biases of the algorithms with even support in time (consistent with Figure 5).

5.3 Effects of Noise

To characterize effects of imaging noise, images of sinusoidal motions were calculated and then degraded by simulated fixed-pattern noise (with energy 50 dB below that of the signal), shot noise (with energy 50 dB below that of the signal), and quantization errors (12-bit resolution). Biases in motion estimates using the periodic-specific 7×7×8 algorithm and using LBC are shown in Figures 10 and 11. The bias functions for the multi-image algorithm share several features across images. First, the bias in the estimate of the x component of amplitude A_x depends on both A_x and A_y. Generally, the biases are smaller when A_x and A_y are small than when A_x or A_y is large. Second, biases for large amplitudes (near $A_x = A_y = 2$) are similar for each of the image sequences. However, biases for small amplitudes differ significantly — being small for the SAR image, larger for the fatigue device, and larger still for the spot. Notably, the standard deviation of the motion estimates (not shown) increases in the same order.

For the spot, the amplitude bias changes little with the amplitude of the displacement. The bias is on the order of 0.01 pixels for the multi-image algorithm and nearly an order of magnitude larger for LBC. Standard deviations for both algorithms (not shown) are approximately 0.01 pixels, independent of the amplitude of the displacement.

For the SAR and fatigue test structures, the amplitude errors of the multi-image algorithm in the presence of noise are qualitatively similar to the amplitude errors in the absence of noise shown in Figure 8. For motions smaller than roughly 1.2 pixels, amplitude errors are smaller than 0.001 pixels.

Bias functions for LBC depend strongly on A_x and only weakly on A_y [26]. Thus, errors in estimating the x component of motion are only weakly affected by the amplitude of motion in the y direction. The bias is nearly constant for $A_x > 0.1$ pixels and decreases rapidly as A_x drops below 0.1 pixels. Except at the highest amplitudes, the bias for LBC is significantly greater than that for the multi-image method.

Figure 12 shows standard deviations for the multi-image algorithm for the fatigue test structure image sequence. Notice that the standard deviation varies little with either A_x or A_y, remaining near 0.001 pixels. Standard deviations of LBC were similar. For the image of the the Galapagos (not shown), standard deviations for both algorithms were 0.006 pixels and uniform.

Phase errors for the multi-image algorithm for the spot sequence are on the order of 10^{-3} radians for all values of A_x and A_y. By contrast, phase errors of LBC become almost as large as 10^{-2} radians. For both the multi-image algorithm and LBC, phase errors for the SAR image and fatigue test structure image are quantitatively similar. The multi-image algorithm phase errors range from 10^{-4} radians to 10^{-2} radians. LBC phase errors remain mostly between 10^{-3} and 10^{-2} radians. Figure 12 shows the multi-image algorithm standard deviation of phase for the fatigue test device image-sequence. The standard deviations range from about 3×10^{-3} to 10^{-2} radians. The standard deviations of phase for the SAR image-sequence are quantitatively similar to those for the fatigue test image-sequence. Standard deviations for phase estimates of the motion of the spot range from a factor of three to a factor of five higher for both algorithms, retaining the same form seen in Figure 12.

6 Discussion

The accuracy of motion measurements from video images is limited not only by physical factors such as noise but also by algorithmic factors, including systematic bias that is present even in the absence of noise. Our goal was to create an algorithm whose inherent errors are smaller than the errors and variations caused by noise. We accomplished that goal. Consider the estimation of sinusoidal motions of a high contrast target (the fatigue test structure). Even the small amounts of imaging noise expected for a high quality, 12-bit, scientific imaging system cause variations in amplitude estimates on the order of 0.001 pixels (Figure 12). By contrast, algorithmic errors for this image sequence are significantly less than 0.001 pixel for motions smaller than a pixel (Figure 8). Thus the bias in the algorithm can be ignored for motions smaller than a pixel.

The spot image sequence was chosen to represent a difficult motion target. The small size of the spot means that fewer than 30 pixels contribute useful information for the motion estimate, even though all 1024 pixels in the spot image sequence contribute to the noise. Imaging noise typical for a high quality camera cause variations in amplitude estimates of the spot on the order of 0.01 pixel (not shown). Errors in the absence of noise for all of the multi-image algorithms are smaller than 0.01 pixel for motion amplitudes

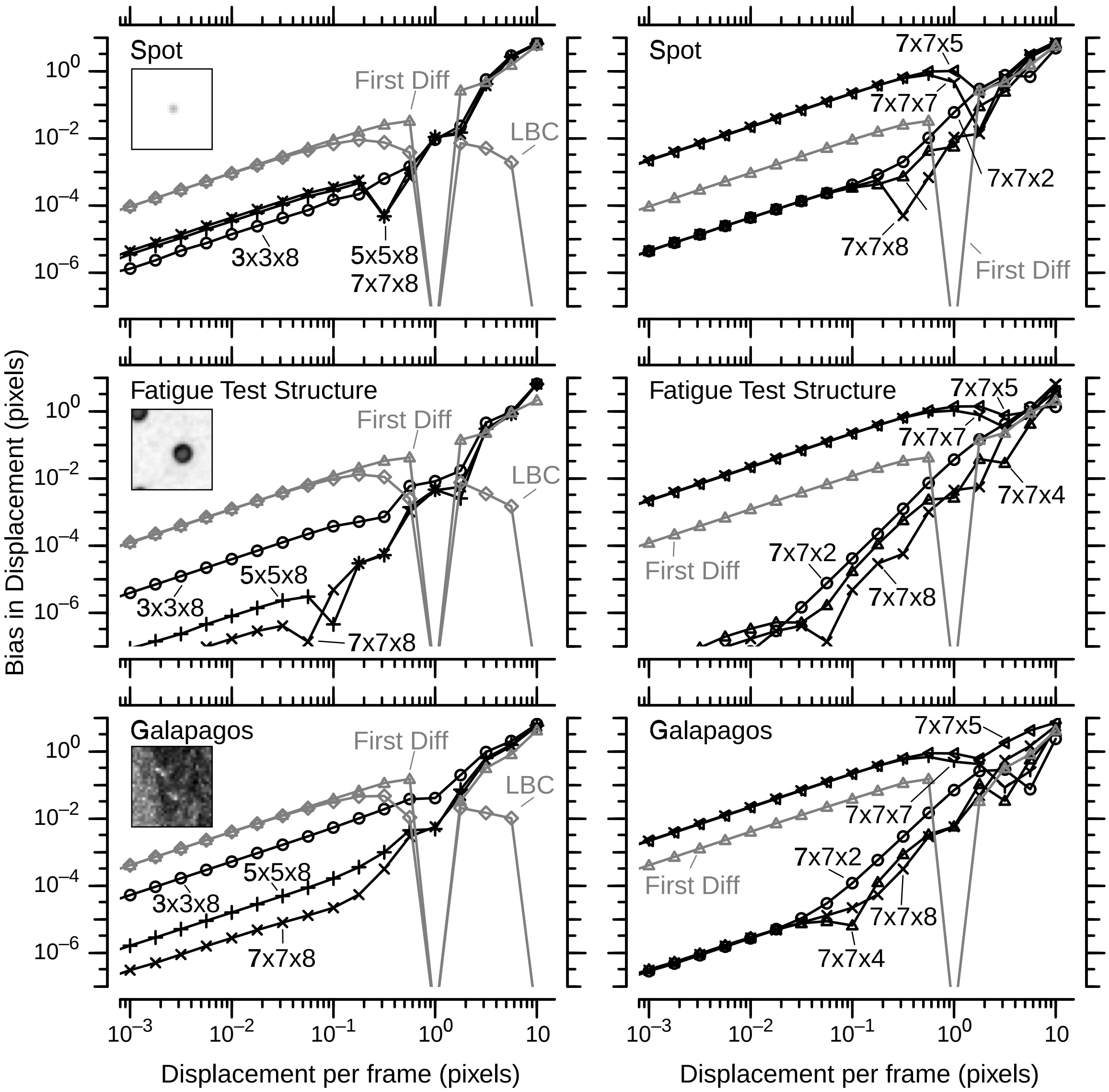

Figure 9. Amplitude dependence of bias in estimates of constant velocity. Constant velocity motions of the spot (upper panels), fatigue test device (middle panels), and Galapagos Island (lower panels) were analyzed for 17 different velocities. For each velocity, 8 images were generated, each displaced from the previous by a constant amount in the x direction (abscissa) and 0 in the y direction. Symbols represent the difference between the estimated displacement per frame and the displacement used to generate the images. The plots on the left compare LBC, the first difference algorithm, and multi-image algorithms with various supports in space. The panels on the right compare multi-image algorithms with various supports in time. The first difference and $7 \times 7 \times 8$ algorithms are shown on both the left and right plots for clarity.

less than 3 pixels (Figure 7). Furthermore, the bias in estimates using the multi-image algorithms are more than an order of magnitude smaller than those for previous algorithms (first differences and LBC) for motion amplitudes smaller than 1 pixel.

As would be expected, in the presence of noise, performance of the multi-image motion estimation algorithms was different for different images. Amplitude biases reached 0.05 pixels for large motions (near 2 pixels) for all of the images (Figure 10). For the multi-image algorithm, bias was smallest for the SAR image and largest for the spot image. This result is consistent with the fact that the signal energy is greatest for the SAR image and least for the spot. In contrast, biases for LBC are not only larger than those for the multi-image method, they are also much less target dependent. The bias in LBC is not significantly improved by the greater amount of signal that is available in the fatigue and SAR image sequences than in the spot image sequence.

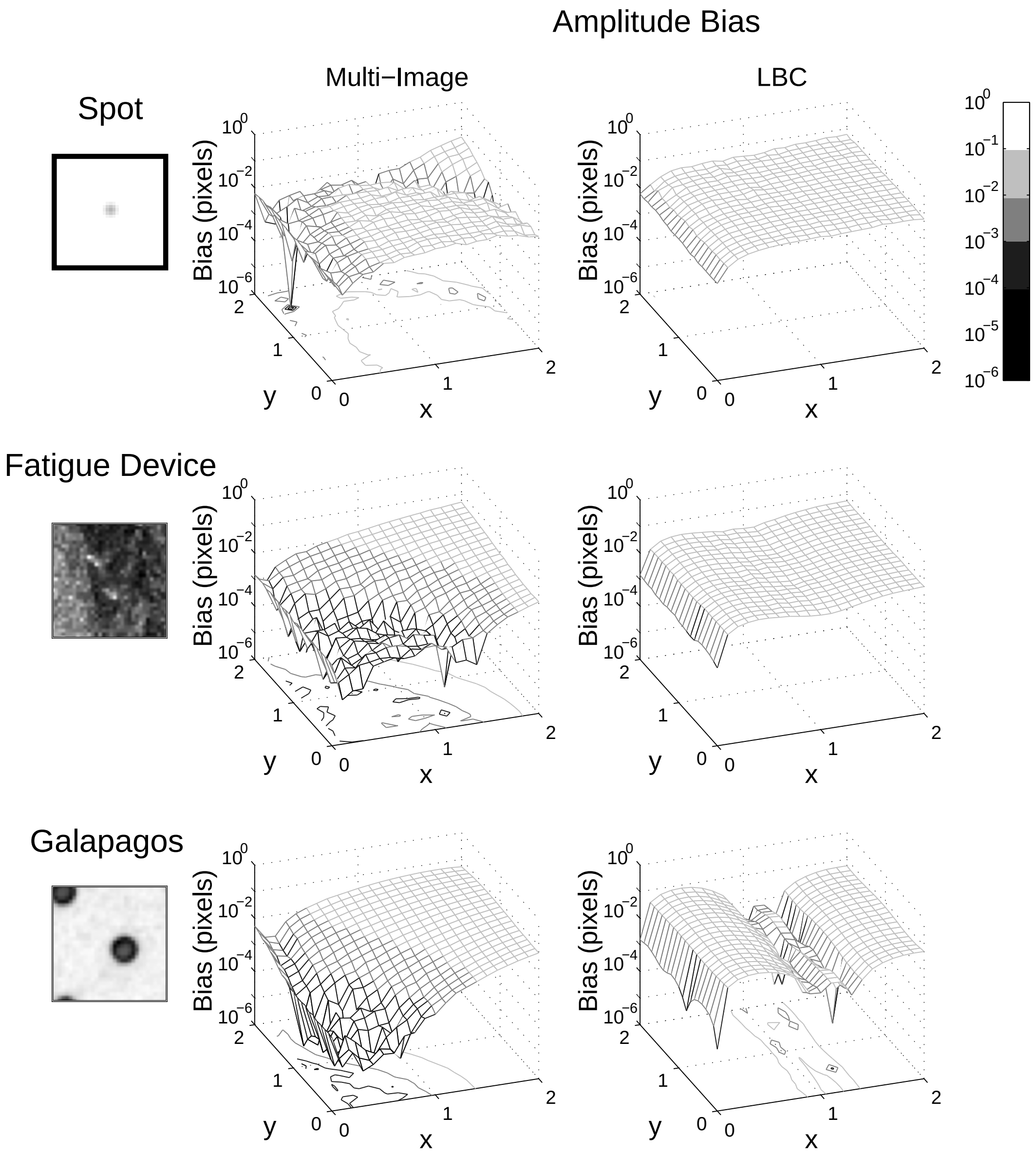

Figure 10. Bias for the estimate of amplitude (A_x) using $7\times7\times8$ algorithm (left) and LBC (right) plotted as a function of driving amplitude, in the presence of -50 dB of shot noise, fixed-pattern noise, and quantization errors. Other aspects of this figure are as described for Figure 8.

6.1 Motions larger than a pixel

The bias in estimating the amplitude of sinusoidal motion becomes large for displacement amplitudes greater than a pixel (Figure 7). One reason for the increase is that while motions may be sinusoidal, the brightness of a pixel does not vary sinusoidally with time. As motions become larger, the energy at high temporal frequencies of the brightness function increases. For motions larger than about a pixel, there is significant energy in temporal frequencies greater than the Nyquist frequency; this energy is aliased to other frequencies, causing errors.

Performance of the multi-image algorithms for large amplitude motions can be improved. For example, if the images are down-sampled, super-pixel motions are converted to sub-pixel motions. The critical part of such a down-sampling strategy is the low-pass filtering, which removes large spatial frequencies. As noted in Section 3.5, these large spatial frequencies cause large temporal frequencies. From this argument, it follows that the low-pass filter used as part of our spatial filters (Figure 1) is not only important to filter out low signal-to-noise regions of the spectrum. It also affects the dynamic range of the motion estimator.

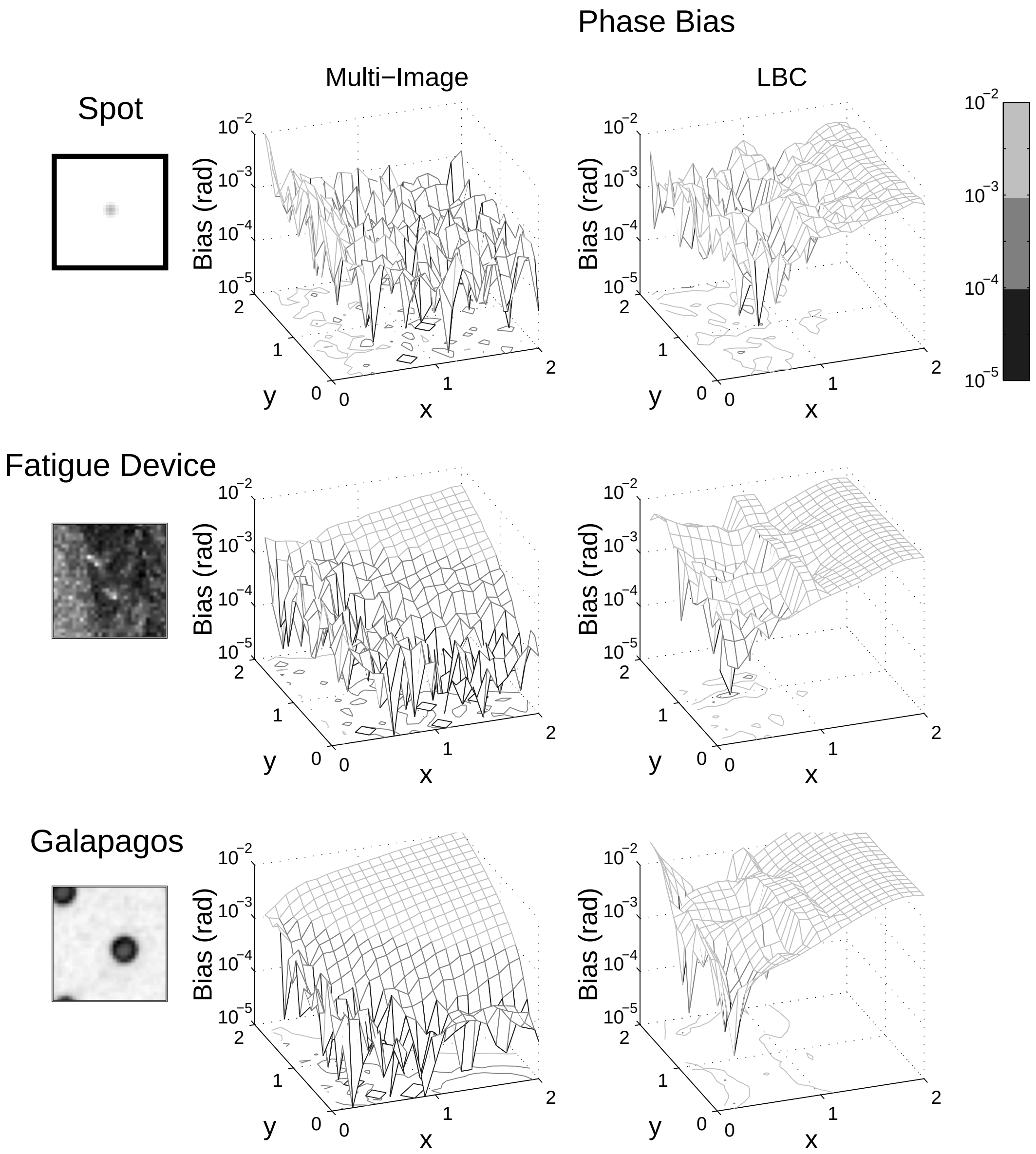

Figure 11. Bias for the estimate of phase using the $7\times7\times8$ algorithm (left) and LBC (right) plotted as a function of driving amplitude, in the presence of -50 dB of shot noise, fixed-pattern noise, and quantization errors. Phase is measured in radians. Other aspects of this figure are as described for Figure 8.

6.2 Volume Support

Filter errors cause motion estimation errors. This is most apparent in Figure 9 where the $7\times7\times5$ and $7\times7\times7$ algorithms perform many orders of magnitude poorer than the $7\times7\times2$, $7\times7\times4$ and $7\times7\times8$ algorithms. Figure 5 shows that the error in the 7-point temporal derivative is much larger than the 8-point derivative filter. Since the $7\times7\times8$ and $7\times7\times7$ algorithms are otherwise the same, we conclude that temporal filter error dominates the $7\times7\times7$ algorithm.

For small motions, errors in spatial filters dominate errors in temporal filters. Figure 9 shows that for motions smaller than about 0.1 pixels per frame, changing from the $7\times7\times2$ to the $7\times7\times4$ to the $7\times7\times8$ algorithm has almost no discernible effects. Conversely, changing from the $3\times3\times8$ to the $5\times5\times8$ or $7\times7\times8$ algorithm has a large effect. We conclude that errors in spatial filters dominate errors in temporal filters for the smallest motions.

For motions larger than about 1 pixel, the performance of the $3\times3\times8$, $5\times5\times8$ and $7\times7\times8$ algorithms converge. Thus, for large motions, spatial filters support is almost unimportant. This can be explained from the discussion in the last section: there is significant energy in high temporal frequencies for large motions. Thus, temporal filters which

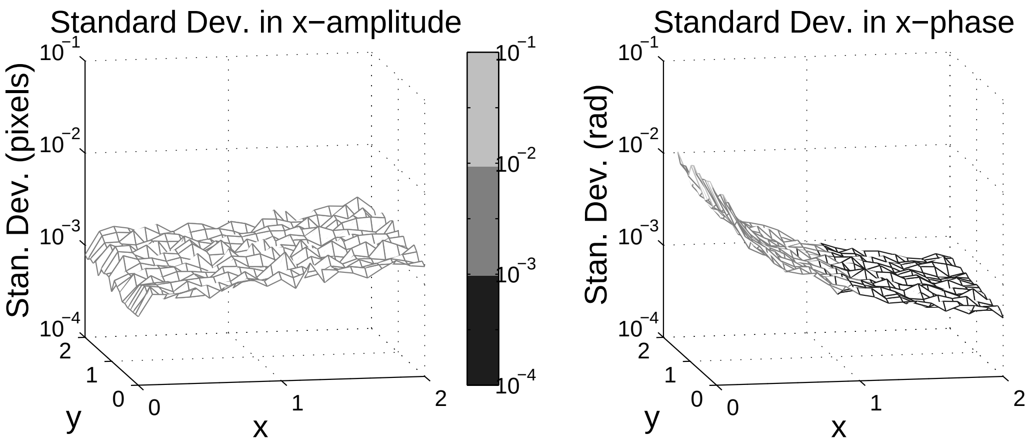

Figure 12. The standard deviation of amplitude and phase estimates for sinusoidal motions of the fatigue test structure using the $5\times5\times8$ algorithm. The standard deviations are computed for the same images used to compute the biases for the fatigue device panels of Figures 10 and 11.

are accurate for high temporal frequencies produce more accurate motion estimates than temporal filters which are accurate only in the smallest part of the spectrum. For the largest motions, temporal frequencies are aliased leading to errors in all temporal filters, regardless of support.

6.3 Comparison of multi-image algorithms with LBC

The bias in LBC estimates is smaller than those for the multi-image algorithms for the largest amplitude motions (Figure 7). This is not surprising, since reducing bias for large amplitude motions was the problem that motivated development of LBC [26]. Also, LBC uses a king stepping approach which allows it to better handle motions larger than a pixel. The bias in LBC estimates is larger than that for the multi-image algorithms for the smallest amplitude motions. In fact, the bias in LBC is nearly equal to that of the first-difference method. It is possible to show that in the limit of small displacements, estimates with LBC approach those based on the first-difference algorithm. Thus for small motions, the bias of LBC is larger than the bias of the multi-image gradient based methods because the new methods use more accurate filters.

6.4 Computational Costs

The motion estimation algorithms that are compared in this paper differ not only in statistical performance but also in computational costs. Implementing the filters directly, the number of computations for multi-image and first-difference algorithms scales as $n_x n_y n_t ABC$ where n_x, n_y , and n_t are the dimensions of the analysis region and A, B, and C are the dimensions of the gradient and interpolation filters in the x-, y- and t- directions. Using symmetries in the filters and normalizing the filters leads to significant computational savings; approximately 90% of the computations are additions, not multiplications. The number of computations for LBC is approximately 4 times the number for the first-difference algorithm. The $3\times3\times8$ algorithm requires slightly more than 2 times as many computations as does LBC. Similar factors for 5×5×8 and 7×7×8 algorithms are 7.3, and 14.2 respectively. On a modern computer, we have found that this increase in computation time is insignificant compared to the time required to read the images from a disk (which is invariant across algorithms). For further speed enhancement, it is also possible to take advantage of the fact that the gradient filters are composed from one dimensional filters to achieve a scaling of $n_x n_y n_t(A+B+C)$.

6.5 Other Image Sequences

The images we examined are not standard ones in the machine vision community. Most of the standard images have been designed to test issues such as occlusion, regions of small gradients, and complex motion fields. We have only addressed accuracy in this paper, not these other complicating issues (although we believe that our methods could be incorporated into other algorithms designed to address these issues).

We briefly report results testing our algorithms with three standard images from Barron et. al: Sinusoid1, Sinusoid2 and the Translating Tree Sequence [34]. Instead of the temporal filters reported in this paper, we use temporal filters that do not compensate for the finite acquisition time of a camera. For the Sinusoid1 image sequence, where $v = (1.585, 0.863)$ pixels/frame, the ($3\times3\times8$, $5\times5\times8$ and $7\times7\times8$) algorithms all performed roughly the same — generating errors less than 0.02 and 0.01 pixels/frame in each direction respectively. These errors are comparable to those reported for the spot, fatigue device, and Galapagos Island images in this paper. For the image sequence Sinusoid2, $v = (1, 1)$, our methods had errors smaller than $1e - 4$ pixels.

For the Translating Tree Sequence, we designed a Lucas and Kanade [35] windowing scheme exactly as described by Barron [34], using our derivative filters. Preliminary tests show our filters yield results comparable to those reported by Barron. Barron effectively uses $15 \times 11 \times 11$ filters in his gradient estimates (larger than the filters used in this study). The Translating Tree Sequence suffers from temporal aliasing along the borders of the objects in the scene. We have not attempted to optimize our filters to compensate for temporal aliasing.

7 Conclusions

We have developed a class of multi-image gradient-based algorithms whose inherent errors for measuring motions are smaller than the errors and variations introduced by noise typical for scientific-grade CCD cameras. The algorithms use filters to accurately estimate derivatives and therefore accurately estimate motions. The methods have more than an order of magnitude less error than the first difference algorithm created by Horn and Schunck [11] and Davis and Freeman's LBC [26].

We introduced a method of using filters to compensate for the effects of the non-zero acquisition time of an imager. The non-zero acquisition time blurs moving objects and would otherwise leads to errors in motion estimates. We also introduced a method for designing spatial derivative and interpolation filters together in order to obtain high accuracy motion estimates with relatively small filter support. Finally, we note that even with only two images, it is possible to improve the error of the first difference algorithm by using higher order spatial filters.

8 Acknowledgments

This work was also supported by a grant from DARPA (F30602-97-2-0106). Samson Timoner is also supported in part by the Fannie and John Hertz Organization. Dennis Freeman is supported in part by the W.M. Keck Career Development Associate Professorship. We acknowledge Dr. C. Quentin Davis who provided comments and participated in insightful conversations.

Appendix 1: Temporal filters

i	$g_t[i] = -g_t[n-i-1]$	$h_t[i] = h_t[n-i-1]$
sinusoid specific, length $n = 8$		
0	-0.149035907898682	-0.0378010678346322
1	0.204171130411112	0.125047021427471
2	-0.408622311811502	-0.267629124130554
3	1.69565453432942	0.680287727944691
general, length $n = 2$		
0	1	0.5
general, length $n = 4$		
0	-0.103968507430399	-0.113916918966049
1	1.31186241952479	0.613878501707367
general, length $n = 5$		
0	-0.942531645436739	0.010688386509402
1	1.28872906527697	-0.0831467273801449
2	0	1.1449196489522
general, length $n = 7$		
0	0.676841149290959	-0.00495209935295687
1	-0.820095721768181	0.0258940146273749
2	1.24079146353296	-0.100860328899541
3	0	1.15983893138018
general, length $n = 8$		
0	-0.00509495368027089	-0.0302094259636097
1	0.0424871505077855	0.12006317028409
2	-0.225014953087821	-0.269034368654828
3	1.49827319026201	0.679172179701493

References

[1] C. Q. Davis and D. M. Freeman. Direct observations of sound-induced motions of the reticular lamina, tectorial membrane, hair bundles, and individual stereocilia. In *Abstracts of the Eighteenth Midwinter Research Meeting*, St. Petersburg Beach, Florida, February 1995. Association for Research in Otolaryngology.

[2] D. M. Freeman and C. Q. Davis. Using video microscopy to characterize micromechanics of biological and manmade micromachines. In *Solid-State Sensor and Actuator Workshop*, pages 161–167. Transducer Research Foundation, Inc., June 1996.

[3] C. Q. Davis and D. M. Freeman. Using a light microscope to measure motions with nanometer accuracy. *Optical Engineering*, 37(4):1299–1304, April 1998.

[4] C. C. Abnet and D. M. Freeman. Deformations of the isolated mouse tectorial membrane produced by oscillatory forces. *Hearing Res.*, 144:29–46, 2000.

[5] M. A. Messerli, G. Danuser, and K. P. Robinson. Pulsatile influxes of H^+, K^+ and Ca^{2+} tag growth pulses of lilium longiflorum pollen tubes. *Journal of Cell Science*, 112(10):1497–1509, May 1999.

[6] D. J. Burns and H. F. Helbig. A system for automatic electrical and optical characterization of micro-

Appendix 2: Spatial Filters

i	$g_t[i] = -g_t[n-i-1]$	$h_t[i] = h_t[n-i-1]$
	ratio minimization, length $n = 3$	
0	0.29464088741622	0.10456437821011
1	0	0.38016162236804
	ratio minimization, length $n = 4$	
0	3.38885794905055	1
1	6.68082443833587	7.42370182447547
	ratio minimization, length $n = 5$	
0	0.57257093939319	0.14965795319880
1	2.75485429075211	1.84711022947292
2	0.00000642906285	3.80645660002629
	ratio minimization, length $n = 7$	
0	0.00196396189835955	$5.20468678432878e-05$
1	0.112139472173319	0.0369446223231693
2	0.365478437146823	0.29028508996955
3	$1.59329214172416e-15$	0.536735013750202
	Parks, length $n = 6$	
0	2.20979793193238	1
1	7.54253708505878	4.65436061439048
2	5.3327391531264	8.72441885207378
	Parks, length $n = 7$	
0	0.570040582359355	0
1	-8.28262952603113	0
2	-22.7375253839524	0
3	0	1
	low pass filter, length $n = 4$	
i	$h[i] = h[3-i]$	
0	0.14962930880927	
1	0.51071846633449	

electromechanic. *Journal of Microelectromechanical Systems*, 8(4):473–482, December 1999.

[7] P. Anandan, M. Irani, R. Kumar, and J. Bergen. Video as an image data source: efficient representations and applications. In *Inter. Conf. on Image Proc., Proceedings, 1995*, volume 1, pages 318–321. IEEE, October 1995.

[8] M. J. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. Computer Vision and Pattern Recognition, CVPR-91*, pages 296–302, 1991.

[9] L. Zelnik-Manor and M. Irani. Multi-frame estimation of planar motion. *IEEE Trans. Pattern Anal. and Machine Intelligence*, 22(10):1105–1116, 2000.

[10] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proc. of the Seventh IEEE Inter. Conf. on Computer Vision, 1999*, volume 1, pages 626–633. IEEE, 1999.

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

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

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

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

[15] A. Hachincha and S. Simon. Subpixel edge detection for precise measurements by a vision system. *Proc. SPIE, Industrial Inspection*, 1010:148–157, 1988.

[16] J. W. Roach and J. K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Trans. Pattern Anal. Machine Intell.*, 2(6):554–562, November 1980.

[17] J. Weng, N. Ahuja, and T. S. Huang. Optimal motion and structure estimation. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(9):864–884, September 1993.

[18] 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.

[19] T. R. Reed. On the computation of optical flow using the 3-d gabor transform. *Multidimensional Systems & Signal Processing*, 9(4):447–452, October 1998.

[20] 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.

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

[22] H.H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. *Computer Vision, Graphics and Image Processing*, 21:85–117, 1983.

[23] Michael Otte and H.H. Nagel. Estimation of optical flow based on higher-order spatiotemporal derivatives in interlaced and non-interlaced image sequences. *Artificial Intelligence*, 78:5–43, 1995.

[24] J. J. Little and A. Verri. Analysis of differential and matching methods for optical flow. In *Proceedings of the Worshop on Visual Motion*, pages 173–80, San Francisco, CA, March 1989. IEEE.

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

[26] C. Q. Davis and D. M. Freeman. Statistics of sub-pixel registration algorithms based on spatio-temporal gradients or block matching. *Optical Engineering*, 37(4):1290–1298, April 1998.

[27] Samson J. Timoner. Subpixel motion estimation from sequences of video images. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1999.

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

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

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

[31] T. W. Parks and J.H. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Transactions on Audio Electroacoustics*, CT-19(4):189–194, Mar 1972.

[32] T.P. Krauss, L. Shure, and J.N. Little. *The Signal Processing Toolbox*. MathWorks, Inc., 24 Prime Park Way, Natick, Mass. 01760, 1993.

[33] Willian H Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vettering. *Numerical Recipes in C*. Cambridge University Press, Cambridge, second edition, 1992.

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

[35] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA IU Workshop*, pages 121–131, 1981.