Quantitative Comparison of Algorithms for Tracking Single Fluorescent Particles

Michael K. Cheezum, William F. Walker, and William H. Guilford
Department of Biomedical Engineering, University of Virginia, Charlottesville, Virginia 22908 USA

ABSTRACT Single particle tracking has seen numerous applications in biophysics, ranging from the diffusion of proteins in cell membranes to the movement of molecular motors. A plethora of computer algorithms have been developed to monitor the sub-pixel displacement of fluorescent objects between successive video frames, and some have been claimed to have "nanometer" resolution. To date, there has been no rigorous comparison of these algorithms under realistic conditions. In this paper, we quantitatively compare specific implementations of four commonly used tracking algorithms: cross-correlation, sum-absolute difference, centroid, and direct Gaussian fit. Images of fluorescent objects ranging in size from point sources to 5 μm were computer generated with known sub-pixel displacements. Realistic noise was added and the above four algorithms were compared for accuracy and precision. We found that cross-correlation is the most accurate algorithm for large particles. However, for point sources, direct Gaussian fit to the intensity distribution is the superior algorithm in terms of both accuracy and precision, and is the most robust at low signal-to-noise. Most significantly, all four algorithms fail as the signal-to-noise ratio approaches 4. We judge direct Gaussian fit to be the best algorithm when tracking single fluorophores, where the signal-to-noise is frequently near 4.

INTRODUCTION

Single particle tracking is the use of computer analysis of video images to follow the sub-micron motion of individual organelles, microspheres, and molecules under microscopic observation. This technique has seen numerous applications in biophysics, including the diffusion of proteins in cell membranes (Ghosh and Webb, 1994), kinesin-driven movement of beads on microtubules (Gelles et al., 1988), and the myosin-driven movement of actin filaments in vitro (Work and Warshaw, 1992). Using this technique, investigators have been able to estimate the diffusion coefficients of proteins in cell membranes and the step displacements generated by "molecular motors." For an excellent overview of the technique, see Saxton and Jacobson (1997).

A plethora of methods have been used for tracking single particles. All include two basic steps. The first is segmentation, in which multiple particles in a field of view are identified and discriminated. Subsequently, an algorithm tracks the particles individually to monitor their displacement between successive video frames. The performance of the tracking algorithm (rather than the segmentation algorithm) defines the fundamental performance limit of the method.

Tracking algorithms used to date have included cross-correlation of subsequent images (Gelles et al., 1988; Kusumi et al., 1993; Guilford and Gore, 1995), calculating the center-of-mass (centroid) of the object of interest (Ghosh and Webb, 1994; Lee et al., 1991), and directly fitting Gaussian curves to the intensity profile (Anderson et al., 1992; Schütz et al., 1997). Some groups have claimed "nanometer" resolution. Unfortunately, there have been no studies quantitatively comparing the efficacy of these algorithms under a variety of conditions. Complicating matters is the fact that many laboratories develop custom-written computer programs for analyzing the data, and incorporate additional thresholds and filters to improve the consistency of their results. This shortcoming is of particular concern since the advent of single fluorophore imaging, in which single fluorescent molecules are observed using intensified video cameras (Sonnleitner et al., 1999; Schütz et al., 1997; Goulian and Simon, 2000). The signal-to-noise ratio (S/N) in these studies can be as low as 3 or 4 (Kubitscheck et al., 2000). Thus, finding the best algorithm for use under these conditions, and knowing its limitations, is vital.

Tracking algorithms may suffer two sorts of errors—determinate and indeterminate. Determinate errors are the result of inaccuracies inherent to the algorithm, systematically biasing the results toward incorrect values. Indeterminate errors, as the name implies, cause the individual measurements to fluctuate randomly, and generally result from sensitivity to underlying noise in the data. Measures of these two error types are colloquially referred to as the "accuracy" and "precision" of the algorithm, respectively.

In general, investigators assess the efficacy of the tracking algorithm by tracking the position of a stationary object, and taking fluctuations in the measured position as a measure of accuracy or precision. Although this technique may be used to set a lower bound on the detectable motion of the particle, to know the accuracy of a tracking algorithm, one must compare the estimated to the actual position of the object being tracked. Unfortunately, there is no way to know the exact position of a real object relative to the imaging array. One group simulated and tracked images of an immobile point source at a fixed S/N ratio, under the

Received for publication 8 January 2001 and in final form 25 June 2001.

Address reprint requests to William H. O. Guilford, Biomedical Engineering, 1105 West Main St., Charlottesville, VA 22903. Tel.: 434-243-2740; Fax: 804-982-3870; E-mail: guilford@virginia.edu.

0006-3495/01/10/2378/11 $2.00

assumption that the S/N fully determines the accuracy of the apparatus and algorithm (Schütz et al., 1997). However, the accuracy and precision of a tracking algorithm are dependent upon the noise, the position of the source relative to the imaging array, the shape and intensity of the object, and the spatial resolution of the imaging system.

The purpose of this study is to quantitatively compare the efficacy of four commonly used tracking algorithms: cross-correlation, sum-absolute difference (SAD), centroid, and direct Gaussian fit. Images of fluorescent particles with diameters both greater than and less than the wavelength of the emitted light (λ) were computer generated, convolved with an appropriate point-spread function, and resampled with known sub-pixel displacement into a lower resolution array representing a video camera. Shot noise was added to give S/N ranging from ∞ to as low as 1.3. One thousand iterations of image pairs were compared with specific implementations of each of the four algorithms for every combination of actual displacement and noise level. The bias error (accuracy) and standard deviation (precision) of the algorithm were determined by comparison to the known displacement. We show that, for the case of a sub-wavelength diameter particle, direct Gaussian fit to the intensity distribution is the superior algorithm in terms of both accuracy and precision. However, all four algorithms fail as the S/N approaches 4. Cross-correlation offers the best performance of the four algorithms when the diameter of the particle is ≥λ. These data have important ramifications for single fluorophore imaging, where the S/N is frequently near 4.

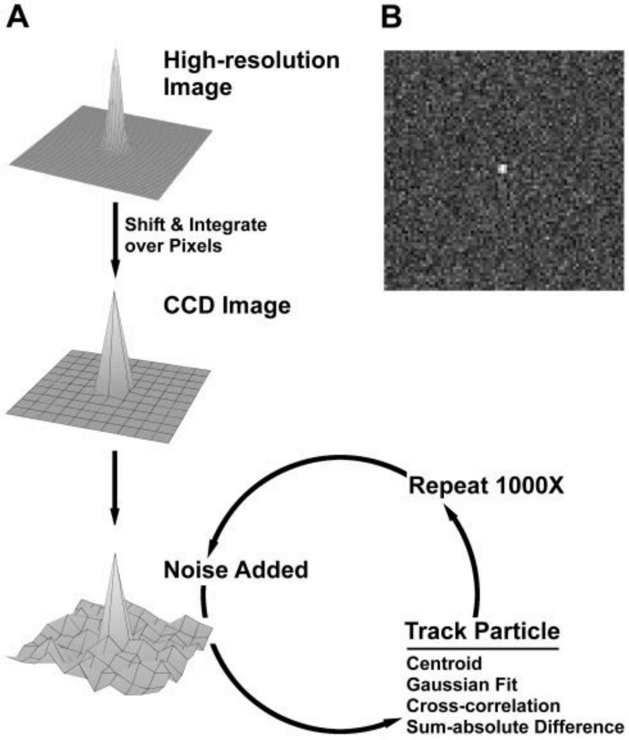

FIGURE 1 (*A*) Schematic illustration of the process of image simulation and testing. A high-resolution image matrix is created that is subsequently shifted relative to and integrated to generate a matrix representing a CCD camera. For each of 1000 iterations of the tracking algorithms, shot noise is applied to the CCD image. (*B*) Generated image of a point source object with shot noise. The mean white level is 40 photoelectrons and the mean background level is 10 photoelectrons. Each pixel is 100 nm in width.

MATERIALS AND METHODS

Image models

To create an accurate model for a fluorescent object imaged with a charge-coupled device (CCD), we first created a high resolution matrix containing the initial, noise-free object function (i.e., image) of the particle to be tracked (Fig. 1 *A*). Each cell in the matrix contains the corresponding intensity of the target in space. Three target sizes were used: a point source ($d \ll \lambda$), a cell-sized object ($d > \lambda$), and an object on the scale of the wavelength of light ($d \approx \lambda$). Assuming a fluorescence emission wavelength of 570 nm tetramethylrhodomine isothiocyanate (TRITC), high-resolution object functions were created of targets sized one pixel (9 nm), 0.5 μm, and 5.0 μm in diameter. The object functions were constructed by assigning a "white level" of 10,000 to each element within the radius of the object, and a "black level" of 0 to all other cells. These levels were later scaled.

Objects viewed through a microscope are distorted by the point-spread function of the objective. To more faithfully model the distribution of intensities in the high-resolution image, we convolved the high-resolution object function with an appropriate point-spread function (PSF),

$$\mathrm{PSF}(r) = \left(\frac{2J_1(ra)}{r}\right)^2, \tag{1}$$

where

$$a = \frac{2\pi \mathrm{NA}}{\lambda}, \tag{2}$$

r is the distance from the origin, NA is the numerical aperture of the objective (1.3), λ is the wavelength of light (570 nm), and J_1 is the Bessel function (Young, 1996). The PSF has a radius (PSF = 0) of approximately 0.27 μm (≈30 pixels in the high-resolution object function). We acquired the convolved image by multiplication of the original image and the PSF in Fourier space, and inverse transformation. No significant magnitudes were found at the edges of the matrix prior to the inverse transform. The point source, represented as one pixel in the high-resolution image, takes on the scaled intensity distribution of the PSF. Larger objects retain their basic original shape but appear diffracted.

CCD image construction

The CCD image was constructed by integrating over rectangular regions corresponding to CCD pixels to form a smaller matrix representing a CCD faceplate (Fig. 1 *A*). The CCD matrix was assumed to cover the same physical dimensions as the high-resolution matrix, but with 1⁄11 the resolution. The factor of eleven was chosen so that the peak intensity of a small object would be centered on a CCD matrix cell rather than a cell boundary. By shifting the high-resolution matrix by one high-resolution element relative to the CCD, and then integrating the underlying cells, we were able to simulate relative displacements of less than one pixel on the CCD. We assumed an objective of 100× magnification, resulting in 0.1 μm/pixel in the CCD matrix, which is typical of values reported in the literature.

A collection of noise-free CCD images with different relative displacements was created by convolving object functions with point spread functions and then integrating. These images were subsequently scaled to give a black level of 10 (assumed to be the photoelectrons/pixel/frame), and a white level between 15 and 1000.

Noise model

Intensifiers for CCD cameras generate shot noise in the image (Ryan et al., 1990). Shot noise is a Poisson process (Papoulis, 1984) where the noise increases as $\sqrt{N}$, N being the number of detected photons or photoelectrons (Ryan et al., 1990). Shot noise was simulated in our experiments by drawing a random value for each pixel from a Poisson distribution of mean N (Press et al., 1997), where N is the level for that particular pixel (see previous paragraph). This value was used as the measured intensity for that pixel.

The result is a realistic image representing a fluorescent particle of known location imaged with an intensified CCD camera (Fig. 1 *B*). Noise was generated independently for every trial of the tracking algorithm.

Tracking Algorithms

Algorithms for tracking the motion of single particles may be divided into two basic categories. The first category is algorithms that estimate the absolute positions of the particle in each image independently. This category includes the center-of-mass, or centroid, algorithm, and direct fits of Gaussian curves to the intensity profile. The second category includes algorithms that estimate the change in position of a particle by comparing an image to one subsequent. This category includes cross-correlation and SAD algorithms.

Centroid

Comparing the center of mass or centroid of two successive images of a particle is a computationally simple and efficient method for estimating the distance an object has moved. For our purposes, an image is a matrix **I** of intensities that contains both an object and a background. Eq. 3 gives the centroid calculation for a single axis.

$$C_{\mathrm{x}} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \sum_{\mathrm{j=1}}^{\mathrm{m}} (x_{\mathrm{i}} \cdot \mathbf{I}_{\mathrm{ij}}) \Bigg/ \sum_{\mathrm{i=1}}^{\mathrm{n}} \sum_{\mathrm{j=1}}^{\mathrm{m}} \mathbf{I}_{\mathrm{ij}}, \tag{3}$$

where x_{i} is the coordinate of a pixel on the x axis, and $\mathbf{I}_{\mathrm{ij}}$ is the intensity of that pixel. To calculate the distance an object has moved, C_{x} is calculated for one image and subtracted from C_{x} for a subsequent image. This equation assumes that the intensities of the object have higher numerical value than the background (not the case in all computer programs). Although this approach is valid for asymmetric particles, the method is especially susceptible to changes in particle shape and orientation between successive images.

It is vital to exclude as much of the image background as possible, lest it strongly bias the centroid calculation to the center of the image. This is accomplished by setting a threshold (expressed here as a fraction of the peak image intensity) that a pixel must exceed to be included in the calculation. There are two methods of handling thresholds. In simple thresholding, values below the threshold level are assumed to be zero, whereas those above threshold are unaltered (in the centroid calculation, this is numerically equivalent to subtracting the threshold value from all pixels, and setting negative values to zero). More commonly, binary thresholding is used, where values below threshold are taken to be zero, whereas those above are taken to be one. We tested both methods.

Gaussian fit

The equation of a two-dimensional (2D) Gaussian curve is of the general form

$$G(x, y) = A \cdot \exp\left[-\frac{(x - x_0)^2 + (y - y_0)^2}{B}\right], \tag{4}$$

where x_0 is the x coordinate of the center of the curve, and A and B are constants. The peak of the point-spread function, and therefore the intensity distribution of a point source, is well approximated by a Gaussian. Thus, directly fitting the above equation to images of subwavelength particles has become a common method of particle tracking (Anderson et al., 1992; Schütz et al., 1997). We fit a 2D Gaussian using a simplex algorithm with a least-squares estimator (Press et al., 1997), allowing the constants A and B to float. As in centroid, independent fitted values of x_0 and y_0 are subtracted to find the displacement between any two images.

Correlation

Correlation (COR) is more computationally intensive than the above techniques (Gelles et al., 1988). This method compares an image (**I**) to a kernel (**K**) of a successive image. **K**, which contains the object being tracked, is shifted relative to **I** in one-pixel increments. For each increment, a correlation value is calculated that describes how well the values in **K** match those of the underlying image, **I**. At the relative shift where **K** and **I** are most similar, one finds a maximum in the correlation matrix, **X**. The cross-correlation between **K** and **I** is given by

$$\mathbf{X}_{\mathrm{x,y}} = \sum_{\mathrm{i=0}}^{\mathrm{n-1}} \sum_{\mathrm{j=0}}^{\mathrm{m-1}} \mathbf{I}_{\mathrm{x+i,y+j}}\{\mathbf{K}_{\mathrm{i,j}}\}, \tag{5}$$

where x and y describe the distance the kernel **K** has moved over the original image **I**. If **K** and **I** are similar except that the object in the image has translated along the x axis by p pixels, then the resulting correlation matrix will have a maximum in cell $\mathbf{X}_{\mathrm{p,0}}$. The kernel dimensions were fixed at 80 × 80 in this study, irrespective of object diameter, to generate unbiased comparisons among different object sizes.

Correlation tends to match the brightest regions of two images rather than the best topographical fit, resulting in errors in some cases. To alleviate this problem, one may use normalized correlation. Each value in the correlation matrix is divided by the root mean square (RMS) of the original image intensities, as shown in Eqs. 6 and 7.

$$XN_{\mathrm{x,y}} = \sum_{\mathrm{i=0}}^{\mathrm{n-1}} \sum_{\mathrm{j=0}}^{\mathrm{m-1}} \frac{\mathbf{I}_{\mathrm{x+i,y+j}}\{\mathbf{K}_{\mathrm{i,j}}\}}{M_{\mathrm{Ix,y}} \cdot M_{\mathrm{K}}}, \tag{6}$$

$$M_{\mathrm{Ix,y}} = \sqrt{\sum_{\mathrm{i=0}}^{\mathrm{n-1}} \sum_{\mathrm{j=0}}^{\mathrm{m-1}} [\mathbf{I}_{\mathrm{x+i,y+j}}]^2} \quad M_{\mathrm{K}} = \sqrt{\sum_{\mathrm{i=0}}^{\mathrm{n-1}} \sum_{\mathrm{j=0}}^{\mathrm{m-1}} [\mathbf{K}_{\mathrm{i,j}}]^2}. \tag{7}$$

n and m are the dimensions of **K**, and M_{K} and $M_{\mathrm{Ix,y}}$ are the RMS values of the kernel and the overlapping portion of the image, respectively.

Normalized covariance is an extension of this concept intended to deal with situations where the image and the kernel have a relative offset in intensity. In this method, one subtracts the mean of kernel **K** from each cell in **K**, and subtracts the mean in **I** in the overlapping region from each cell in **I**. Eq. 10 describes the normalized covariance calculation.

$$\bar{\mathbf{I}}_{x,y} = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\mathbf{I}_{x+i,y+j}}{n\cdot m}, \tag{8}$$

$$\bar{\mathbf{K}} = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\mathbf{K}_{i,j}}{n\cdot m}, \tag{9}$$

$$VN_{x,y} = \sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\frac{[\mathbf{I}_{x+i,y+j} - \bar{\mathbf{I}}_{x,y}]\{\mathbf{K}_{i,j} - \bar{\mathbf{K}}\}}{M_{Ix,y}\cdot M_{K}}. \tag{10}$$

Sum-absolute difference

The SAD method determines the translation of **I** relative to **K** that minimizes the sum of absolute differences between the overlapping pixels,

$$\mathrm{SAD}_{x,y} = \sum_{i=0}^{k-1}\sum_{j=0}^{k-1}\left|\mathbf{I}_{x+i,y+j} - \mathbf{K}_{i,j}\right|. \tag{11}$$

In contrast to the algorithms above, if the object in the image has translated along the x axis by N pixels, then the resulting SAD matrix will have a minimum in cell $\mathbf{S}_{-N,0}$. Although this algorithm has never been used for tracking fluorescent particles, it is a standard algorithm for tracking the motion of features in medical imaging (Bohs et al., 1993). We included tests of this algorithm for completeness.

Interpolation methods

The centroid and Gaussian fit methods inherently return sub-pixel estimates of distance moved because the position is calculated as an average over a set of coordinates. However, methods that compare subsequent images return discrete matrices, and thus offer only whole-pixel estimates. To achieve sub-pixel resolution, the correlation, covariance, or sum-absolute difference matrices must be interpolated to find the maximum or minimum.

The data in the SAD or correlation matrices form quasi-paraboloid meshes, where the x, y, and z-coordinates are x and y distances moved, and the corresponding sum-absolute difference or correlation values, respectively. Three functions were used to interpolate our data: paraboloid, consinusoid, and Gaussian. We used closed form solutions to fit paraboloids and cosinusoids to the maximum and four nearest neighbors in the correlation matrix (or minimum in the SAD matrix). Briefly, a parabaloid defining a 2D surface is described by

$$z = a + bx + cy + dx^2 + ey^2. \tag{12}$$

x and y designate the coordinates within the correlation matrix. The coefficients in Eq. 12 are obtained from the cell containing the peak of the correlation, $z(x_0, y_0)$, and the four immediately surrounding points on the x and y axes:

$$\begin{aligned}
a &= z(x_0,y_0)\\
b &= \tfrac{1}{2}\,(z(x_1, y_0) - z(x_{-1}, y_0))\\
c &= \tfrac{1}{2}\,(z(x_0, y_1) - z(x_0, y_{-1}))\\
d &= -z(x_0, y_0) + \tfrac{1}{2}z(x_1, y_0) + \tfrac{1}{2}z(x_{-1}, y_0)\\
e &= -z(x_0, y_0) + \tfrac{1}{2}z(x_0, y_1) + \tfrac{1}{2}z(x_0, y_{-1})
\end{aligned} \tag{13}$$

x and y are solved directly using the equations

$$x_{\max} = b/2d \quad y_{\max} = c/2e. \tag{14}$$

Cosinusoidal interpolation was accomplished as in deJong et al. (1990). The peak of the correlation function relative to the reference image is given by

$$\delta_{\max} = -\theta/\omega, \tag{15}$$

where ω is the angular frequency and θ the phase of a cosinusoid, given by the equations,

$$\omega = \cos^{-1}\left(\frac{z(x_0, y_0) - z(x_{-1}, y_0)}{2z(x_1, y_0)}\right), \tag{16}$$

$$\theta = \tan^{-1}\left(\frac{z(x_0, y_0) - z(x_{-1}, y_0)}{2z(x_1, y_0)\cdot \sin\omega}\right). \tag{17}$$

The notation designating the cells in the COR/SAD matrix is the same as in the description of parabolic interpolation, with $z(x_0, y_0)$ designated as the peak intensity in the matrix. The above implementation is for one-dimensional fitting only.

Finally, one can use a Gaussian to fit the peak in the COR/SAD matrix. We used a simplex algorithm (Press et al., 1997) with a least-squares estimator to fit a Gaussian to the maximum and four nearest neighbors.

Implementation

As previously indicated, the background level was held constant at 10 photoelectrons/pixel. Altering the maximum photoelectron count for the object (white level) between 15 and 1000 photoelectrons/pixel changed the signal-to-noise ratio of the image between 1.3 and 31.2. We also considered the deterministic case of $S/N = \infty$ (no noise). The original high-resolution matrix was displaced by one or more high-resolution cells over the CCD matrix, resulting in known displacements of the object in multiples of 1⁄11 CCD pixels, or the equivalent of ≈9 nm.

One thousand trials of each algorithm were performed for each condition of distance moved, S/N, and threshold level (for centroid) to obtain a bias (B) and standard deviation σ for the condition,

$$B = \langle a - \hat{a}\rangle \quad \sigma = \langle (a - \langle a\rangle)^2\rangle^{1/2}. \tag{18}$$

Error for the bias is equal to σ for the same condition. Error for σ was determined by calculating σ for 10 sets of 100 successive iterations, and taking the standard deviation of the resulting independent estimates. The original high-resolution images were created using MatLab on an IBM RS6000 43P workstation. All subsequent numerical experiments were conducted on a Intel Pentium class microcomputer (Dell Optiplex) using code written in Borland C++. Graphs were generated using SigmaPlot (SPSS Inc.).

RESULTS

Thresholding in centroid

For point source objects, the precision and accuracy of the centroid algorithm were critically dependent upon the threshold level applied to the image. As expected, lower thresholds give poorer results at low S/N. With no threshold applied to the image, bias remains high at all S/N levels (Fig. 2 *A*). This is because all the background pixels in the image are included in the center-of-mass calculation, bias-

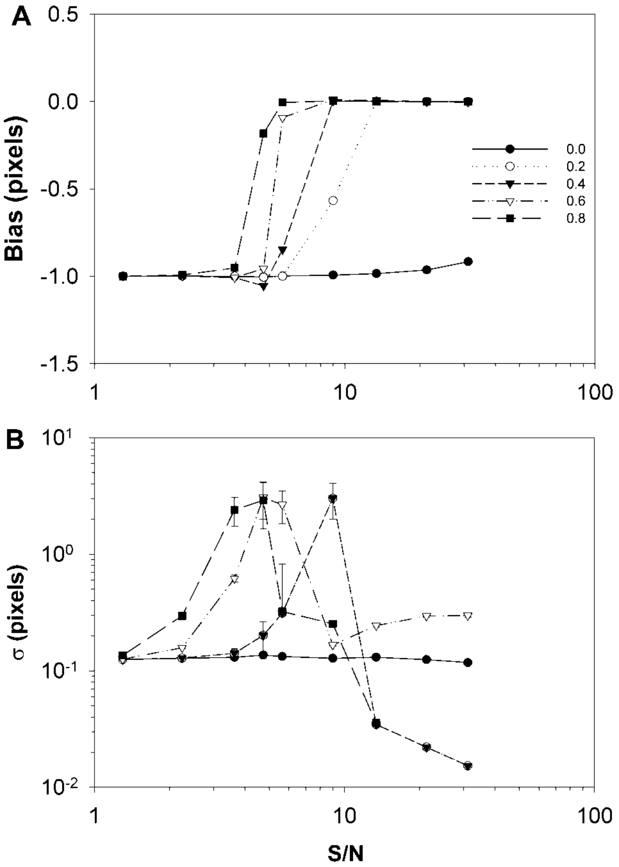

FIGURE 2 (*A*) Bias versus S/N and threshold in centroid for a point source target. Actual displacement was 1 pixel. Negative bias is toward the reference (unmoved) image. Errors are represented in the lower panel. (*B*) Standard deviation versus S/N and threshold in centroid. Error bars indicate one standard deviation. Threshold values are indicated in the graphs. Standard deviation values of 0 are omitted.

ing the result toward the geometric center of the image. Thus, the bias is always roughly equal in magnitude to the actual displacement. Increasing the threshold (limiting the pixels included in the centroid calculation to only the brightest) results in less bias at lower S/N. However, no significant improvement is seen at any threshold level below 3.6 S/N.

The relationship between σ and S/N is more complex (Fig. 2 *B*). At 1.3 S/N, σ returned by all threshold levels are low and roughly equal ($\approx$0.1 pixels). At such a low S/N, even a moderately high threshold (expressed as a fraction of the peak intensity in the image) includes many background pixels in the centroid calculation. As a result, the background pixels in the image bias the result toward the geometric center of the image, giving a consistent result in each trial. At slightly higher S/N, many fewer background pixels are included in the centroid calculation, which introduces tremendous variability into the position estimates, even though they become more accurate. This is reflected in a peak in the σ versus S/N plot approaching 3 pixels for an 80×80 region of interest, typically at $\approx$4 S/N. Increasing the threshold results in a shift of this peak to lower S/N, suggesting that higher thresholds are preferable. However, at a threshold of 0.8, the σ drops to zero above 13.4 S/N. Indeed, at a S/N of 31 and thresholds 0.8 and 0.2, we find $\sigma = 0$ and $B = -$(actual distance moved) for certain sub-pixel displacements (not shown). These "discontinuities" in the trend of σ versus S/N result from a strong bias to the nearest pixel, giving the same result in each trial. Obviously, sub-pixel performance is severely compromised at these threshold levels. A threshold of 0.4 returned the lowest bias and σ at moderate S/N while maintaining sub-pixel performance ($\sigma \neq 0$). Thus, a threshold of 0.4 was used for the remaining comparisons.

We compared two methods for thresholding, described in Methods. We found that binary thresholding—converting pixels above threshold to 1, and those below threshold to 0—resulted in higher biases and discontinuous relationships among σ, S/N, and actual displacement (data not shown). Simple thresholding, where all values below the threshold were set to 0, was used through the remainder of these studies.

Normalized correlation and covariance

Normalized correlation and normalized covariance performed similarly under all conditions to non-normalized cross correlation, except at S/N $<$ 3, where the differences in performance were trivial. Thus, only non-normalized correlation will be considered for the remainder of the manuscript.

Comparison of algorithms

Bias as a function of distance moved

Bias was periodic for centroid, COR, and SAD as a function of actual distance moved, with a period of 1 pixel (Fig. 3 *A*). Qualitatively, bias above one pixel of actual movement did not differ from bias below one pixel. For example, bias for a given algorithm will be the same at 0.3, 1.3, 2.3, . . . pixels actual displacement, except at very low S/N where the algorithms cease to function. At these "limiting" S/N, bias will often scale proportionately to the actual displacement. However, for simplicity, we will only consider actual displacements over the range of 0–1 pixel.

Only the highest S/N is shown in Fig. 3. Although the magnitude of the bias increases with decreased S/N, the results are readily generalized to lower S/N. For all algorithms, bias is approximately zero at multiples of 0.5 pixels. For COR and SAD, bias is toward the nearest whole pixel with a maximum bias at ~0.3 pixels actual distance moved. At high S/N (31.2 in Fig. 3 *A*), bias reaches a maximum of $\approx$0.1 pixel in the case of SAD. COR was nearly an order of

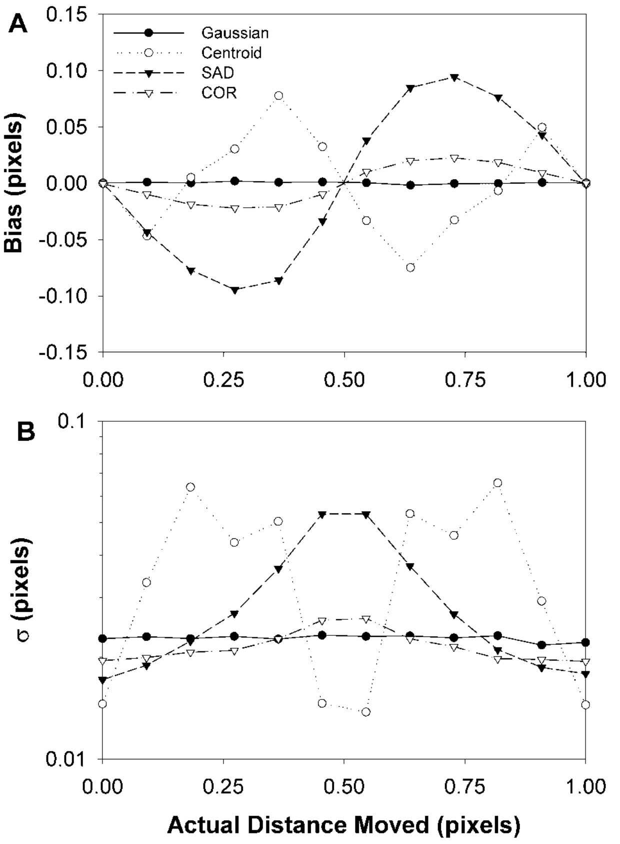

FIGURE 3 (*A*) Bias versus actual distance moved for a point source target: ●, Gaussian fit; ○, centroid; ▼, sum-absolute difference; and ▽, cross-correlation. S/N was 31.2. Note that bias in direct Gaussian fit is largely independent of distance moved. Errors are represented in the lower panel. (*B*) Standard deviation versus actual distance moved. Note that variance in direct Gaussian fit is largely independent of distance moved. Error bars fall within the symbols.

magnitude less biased under the same conditions. These values are not significantly different from the performance at infinite S/N (data not shown). Peak bias was reduced by 25% by using cosinusoidal interpolation rather than parabolic, so all the remaining data use cosinusoidal interpolation. In contrast, centroid was biased toward the midpoint between adjacent pixels, reaching a maximum bias of $\approx$0.08 pixels at high S/N. Direct Gaussian fit to the image was effectively unbiased and independent of actual displacement.

Precision as a function of actual distance moved

Standard deviation at the highest S/N was independent of actual distance moved for direct Gaussian fit at $\approx$0.02 pixels (Fig. 3 *B*). In contrast, σ returned by the three other algorithms was highly dependent on actual distance moved. For SAD and COR, σ was lowest at whole pixel displacements, and maximal at the midpoint between pixels. Centroid returned the lowest σ at whole and half pixel movements, and highest at intermediate motions. COR returned the lowest σ overall, whereas SAD and centroid returned the highest. σ did not change significantly when cosinusoidal interpolation was used instead of parabolic.

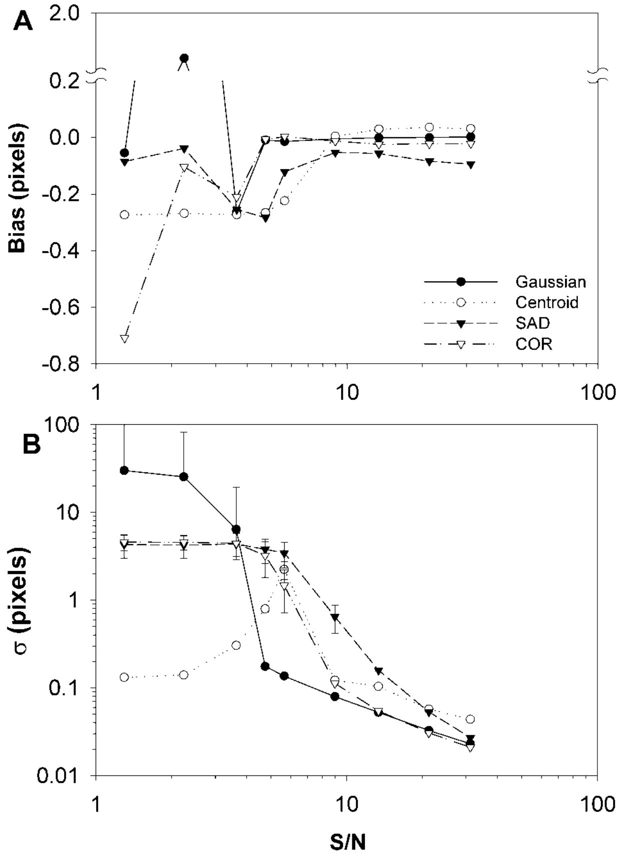

FIGURE 4 (*A*) Bias versus S/N for a point source target: ●, Gaussian fit; ○, centroid; ▼, sum-absolute difference; and ▽, cross-correlation. Actual distance moved was 0.27 pixels. Errors are given in the lower panel. (*B*) Standard deviation versus S/N. Error bars indicate one standard deviation.

Bias as a function of S/N

Figure 4 shows bias and σ as a function of S/N for 0.27 pixels moved. An actual displacement of 0.27 pixels was chosen because it results in the greatest overall bias (see Fig. 3). However, the results generalize readily to all displacements. At low S/N, the relationship between bias and S/N is relatively complex (Fig. 4 *A*). At S/N $= 1.3$, the two comparative algorithms, COR and SAD, choose nearly random locations within the matrix. Biases in this case may be randomly high or low in any given round of simulations, but fall toward zero at S/N ≈ 3.6. Similarly, Gaussian fit is free to return values outside the actual image boundaries, which can result in very high biases at low S/N.

In contrast, centroid begins with a high bias at limiting S/N, and falls monotonically toward zero. Centroid tends to

TABLE 1 Approximate S/N at which bias and σ drop below 0.1 pixels and 1.0 pixels, respectively, for various algorithms

Algorithm	$S/N_{0.1\text{bias}}$	$S/N_{1.0\sigma}$
Gaussian fit	4.2	4.0
Centroid	7.8	6.6
Sum-absolute difference	6.9	8.1
Cross-correlation	4.2	6.3

return the center of the image matrix at limiting S/N, and so the bias is approximately equal to the actual displacement under these conditions.

As an arbitrary gauge of performance, we determined the S/N at which the bias drops below 0.1 pixels on the descending limb of the bias versus S/N curve (Table 1). Gaussian fit and COR performed best, returning <0.1 pixels bias at S/N > 4.2. In contrast, centroid did not return <0.1 pixels bias until S/N > 7.8. The method of interpolation had a negligible effect on the S/N at which bias fell below 0.1 pixels in COR or SAD.

Precision as a function of S/N

For all but one algorithm (centroid), σ decreased monotonically as S/N increased (Fig. 4 *B*). Standard deviation at 1.3 S/N for COR and SAD are equal, and fully explained as the variance one would predict by choosing random locations within the COR or SAD matrix. Gaussian fit at low S/N exhibits very high σ, as the curve-fitting algorithm may return random values that fall outside the image. In contrast, centroid returned low σ at 1.3 S/N, indicating that the algorithm reproducibly finds the center of the image rather than the center of the object. For centroid, σ increases as S/N rises to 5.6 S/N, and then falls. Once again, as an arbitrary gauge of performance, we determined the S/N at which σ drops below 1 pixel on the descending limb of the σ versus S/N curve (Table 1). Gaussian fit performed the best, reaching $\sigma < 1$ pixel at S/N $\approx$ 4.

Large objects

When the object is much larger than the wavelength of light ($d > \lambda$), the profile of pixel intensities is no longer described by a Gaussian. Gaussian fit has therefore been omitted from this portion of the analysis. Among the three remaining algorithms, correlation still performed the best in terms of bias error (Fig. 5 *A*). Bias versus actual distance moved is reminiscent of that for $d \ll \lambda$ (Fig. 3 *A*). However, both COR and SAD remained at <0.1 pixels bias at all S/N tested. Centroid, in contrast, did not fall below 0.1 pixels bias until S/N > 4.7. Overall, the bias for each of the algorithms at low S/N was improved relative to tracking the point-spread object, whereas, at high S/N, improvements were trivial.

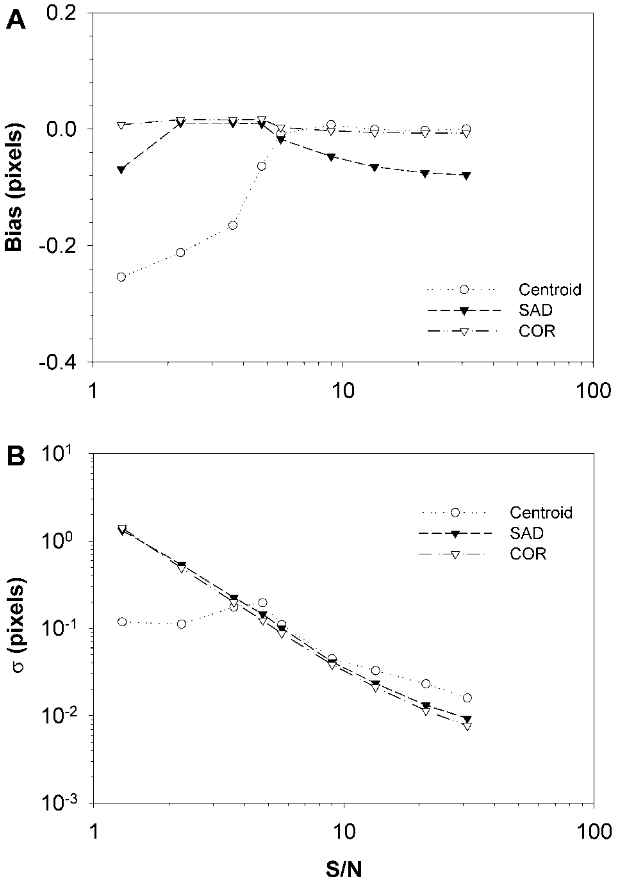

FIGURE 5 (*A*) Bias versus S/N for a 5-μm target: ○, centroid; ▼, sum-absolute difference; and ▽, cross-correlation. Errors are given in the lower panel. Actual distance moved was 0.27 pixels. The scaling is the same as in Fig. 4 *A*. (*B*) Standard deviation versus S/N.

For SAD and COR, σ was highest at 0.5 pixels moved, whereas centroid displayed uniform σ with respect to distance moved at high S/N (data not shown). σ was approximately one order of magnitude lower at each S/N compared to $d \ll \lambda$ (Fig. 5 *B*). This is expected because a larger target has a higher total energy, making motion estimation less susceptible to noise. The results at $d \approx \lambda$ were qualitatively similar and quantitatively intermediate to those at $d \ll \lambda$ and $d > \lambda$, and will therefore not be discussed.

DISCUSSION

We have demonstrated significant differences between common implementations of particle tracking algorithms both in terms of their accuracy and precision. For particles that are smaller than one wavelength in diameter, direct fit of a Gaussian curve to the intensity profile may be judged the best method by several criteria. This method achieves a given precision at lower S/N compared to the other algorithms tested. Further, both the bias and the variance are independent of the actual distance moved—a characteristic unique among the algorithms tested. When direct Gaussian fit cannot be used, such as when the particle is much larger than one wavelength in diameter, cross-correlation appears to be the best choice.

Why does Gaussian fit operate with less bias than COR? Gaussian fit is similar to COR, in that a kernel (function) is matched to an underlying matrix containing pixel intensities. However, in Gaussian fit, the kernel is perfect—that is, noise-free and known to fit the intensity distribution across the object. Thus, Gaussian fit operates at a lower effective S/N than COR (see Walker and Trahey, 1994). Alternatively, when COR is used with a second or subsequent image as a kernel, one would expect the error in the estimates to increase by at least $\sqrt{2}$ (Walker and Trahey, 1994). Although this is often a trivial increase, it becomes important when S/N is limiting, as we have demonstrated. One might also generalize that, when tracking point source objects, COR cannot match the precision of Gaussian fits, whatever the exact implementation of the algorithm.

Our data comparing COR, normalized COR, and SAD do not agree with a related study by Friemel et al. (1995). These same three algorithms were used to track synthetic patterns representing speckle in ultrasound images. Friemel found that normalized COR and SAD performed similarly at all S/N, judged in terms of σ, whereas non-normalized COR performed significantly worse and converged with the others only as S/N became very high. However, there are two significant differences between our two studies. First, speckle patterns are continuous and relatively uniform. The net effect may be to increase the effective spatial sampling density per object relative to a discrete object. Indeed, tracking 5-μm objects increases sampling density per object relative to a point source, and we see that SAD and COR converge in their performance (Fig. 5). Second, Friemel dealt with additive noise rather than shot noise. Whatever the exact source of the discrepancies, the differences between these two studies highlight the importance of understanding tracking algorithm performance in each unique imaging modality.

Limiting S/N

One of the most important indices of algorithm performance is the limiting S/N. That is, what minimum S/N is necessary for the algorithm to function? We have rather arbitrarily defined "limiting" as that S/N at which the worst-case bias falls to 0.1 pixels and the standard deviation to 1 pixel at 0.27 actual pixels moved. Although the choice of these limits is completely arbitrary, they do have some practical relevance. At a standard deviation of 1 pixel, or 1 pixel RMS, 67% of the time the algorithm can be expected to return the correct value to within 1 pixel. This then is a reasonable gauge of the S/N needed for 1-pixel precision. Although the choice of 0.1 pixel's worst-case bias would seem to be an order of magnitude a more stringent requirement for accuracy, both requirements give conveniently similar estimates of the minimum required S/N.

By these definitions, we can estimate that the limiting S/N for direct Gaussian fit is approximately 4. For comparison, limiting S/N for cross-correlation, SAD, and centroid are very roughly 6.3, 8.1, and 6.6, respectively. Quantitatively, this agrees well with predicted values from Walker and Trahey (1994). None of these algorithms should be expected to return believable results if the S/N is less than the corresponding limit.

Properly estimating the S/N is critical to application of these guidelines, but is most often calculated incorrectly. Shot noise is proportional to the square root of the number of photoelectrons ($\sigma \propto \sqrt{N}$). As a result, the noise level is higher in pixels containing a fluorescent object than in the background, even though the noise may be more visually apparent in the background. Failure to account for this relationship may lead to very large errors in estimated S/N. For example, assume a fluorescent object on an imperfectly dark background. Assume that the mean photoelectrons/frame/pixel detected for the object is 144, while, for background, it is 16. The mean RMS noise in these two regions of the image is expected to be $\sqrt{144}$ and $\sqrt{16}$, or 12 and 4, respectively. S/N is calculated as the difference in mean intensity between the object (I_o) and background (I_b), divided by a representative noise level (σ),

$$\mathrm{S/N} = (I_o - I_b)/\sigma. \tag{19}$$

In our example, choosing for σ the background RMS (σ_b) (as in Sonnleitner et al., 1999) would yield a S/N estimate of (144 − 16)/4, or 32. However, the RMS variation in the background is an inappropriate choice for σ, because it is the lowest noise region in the image. Thus, the noise over the object (σ_o) yields a more accurate estimate of S/N ($\approx$11). This was the method used to determine the S/N in this study. A more conservative estimate can be obtained by propagating the noise error, as $\sqrt{\sigma_b^2 + \sigma_o^2}$ (Kubitscheck et al., 2000), which, in this example, yields S/N $\approx$ 10.

Implications for single molecule imaging

These limits are of critical importance for the emerging field of single fluorophore imaging. Whether detected using an intensified CCD camera (Goulian and Simon, 2000; Kubitscheck et al., 2000) or a spot-confocal system (Warshaw et al., 1998), the S/N when detecting single fluorescent molecules in real time is generally limited to approximately 3–4. Although a S/N of 4.0 is just barely acceptable by our criteria, a drop in S/N from 4 to 3 increases bias by approximately four-fold, and variance by two orders of magnitude when Gaussian fit is used! Thus, taking steps to maximize S/N is crucial when particle tracking is to be used with single fluorophores.

Claims of sub-pixel resolution

Claims of nanometer or even sub-pixel resolution should be viewed with some skepticism. It is true that, given our conditions and at the highest S/N examined (31.2), direct Gaussian fit has a predicted mean accuracy (under our assumed conditions) of 1 nm and a precision of ≈2 nm. However, all these algorithms degrade rapidly in performance below 10 S/N. Gaussian fit, for example, degrades to 10-nm precision at ≈9 S/N, and to only 100-nm precision (1 pixel) at 4.0 S/N. These values are in reasonable agreement with Kubitscheck et al. (2000). Reasonable precision can be expected from all four algorithms at high S/N, though predicted biases of several nanometers persist for all but Gaussian fit. Even here, one should consider additional sources of noise and inaccuracy that will undoubtedly decrease accuracy and precision in all algorithms (see Limitations).

Implicit in these statements is that the tracking algorithm has been properly optimized. The centroid algorithm, for example, has seen many variations. The size of the image kernel, the threshold level, the option of converting the image to binary before thresholding, and image averaging (Goulian and Simon, 2000) can all affect the accuracy and precision of the algorithm. Assume, for example, that one used centroid, and chose the threshold level to give the lowest variability in the position of a fixed object. Although this is certainly an intuitive approach, the likely cause of the low variability is that the investigator has chosen a threshold that biases the algorithm consistently to the center of the nearest pixel. Though the variability in estimated positions might imply nanometer precision, the accuracy would actually be on the order of 1 pixel (typically 100 nm).

It is common to track the position of a fixed particle and use the variance of the positions as an indicator of both precision and accuracy. However, such empirical tests of tracking algorithm efficacy are lacking in two respects. First, one does not know if the particle is truly motionless with respect to the imaging system. Second and more importantly, one does not know the actual position of the particle relative to the elements in the imaging system. Without this knowledge, one cannot estimate the bias in the tracking algorithm, which indicates the true accuracy of the measurement. Claims of sub-pixel accuracy and precision (or even whole-pixel precision in very low contrast images) must be demonstrated in numerical simulations or more rigorous experiments than those performed to date. These might include tracking of microspheres simultaneously through video and an alternative, higher resolution technique, such as back-focal plane interferometry (Allersma et al., 1998).

Origins of bias

At very low S/N, very large biases can result. At low S/N, centroid tends to weight toward the geometric center of the region of interest, regardless of where the tracked feature resides. Gaussian fits to a noisy image can return values completely outside the image unless artificially constrained. COR and SAD may return random whole-pixel estimates of location within the COR/SAD matrix. Further, the interpolation function may give a value that lies outside the matrix boundaries. These effects can result in randomly high or low apparent biases on any given round of simulations.

At moderate-to-∞ S/N, centroid, SAD, and COR can still be biased, though the underlying cause is not as apparent. Centroid is biased because the centroid of a continuous function need not equal the centroid of a sampled function (like an image). It has been proposed that sub-pixel bias in COR and SAD results from the failure of the interpolation algorithm to accurately account for the shape of the peak in the COR/SAD matrix (deJong, 1990). Indeed, we have no a priori knowledge of the shape of the peak. The fact that cosinusoidal interpolation is less biased than parabolic anecdotally supports this conclusion, as does visual inspection of parabolic and cosinusoidal fits to our COR and SAD matrices (not shown). Direct Gaussian fit to the image is presumably unbiased because it accurately reflects the shape of the peak of the PSF.

To eliminate assumptions about the shape of the peak in the correlation matrix, others have estimated the centroid of the peak (Gelles et al., 1988) instead of resorting to interpolation. This again requires the use of thresholds to isolate the peak in the centroid calculation—a parameter whose effects must be carefully assessed. Further, the correlation matrix is a sampled function, so it is probable that the centroid of this matrix will be biased (see previous paragraph). However, we did not extensively test this particular implementation of correlation.

Limitations

This study focused on the simplest case of particle tracking—a point-source. The situation becomes much more complex when the "particle" becomes greater in dimension than λ, and when the particle is no longer symmetric. One example is the tracking of cell movements through laminar flow chambers (Smith et al., 1999). Although our data on fluorescent objects $>\lambda$ in diameter suggests excellent behavior both in terms of bias and σ, we did not simulate the entire range of variables expected of a real, moving cell. Cells are not of predefined shape, can change apparent shape over short time spans, and are not ordinarily uniformly fluorescent.

For the centroid calculations, we expressed the threshold as a fraction of the noise-free peak image intensity, and it was therefore constant for any given S/N. However, should thresholds be thus applied to real images, the threshold value for any given image will vary due to variations in the peak intensity, and this may randomly vary the bias. If the background level of a series of images is relatively constant, it may be adequate to set an absolute intensity for the threshold. Otherwise, setting a threshold relative to the mean pixel intensity may be a better choice.

Another example of single particle tracking applied to relatively large, asymmetric particles is the tracking of myosin-driven movement of fluorescent actin filaments in the in vitro motility assay (Work and Warshaw, 1992). Measuring the frame-to-frame velocity of actin filaments has proven a very fruitful technique. Although a number of techniques have been developed for tracking actin filaments, centroid is the most common underlying algorithm. This is cause for concern, because the S/N in these experiments, which effectively involve single molecule imaging, is ≈5, and because the actin filament can change shape and orientation as it progresses randomly across the myosin-coated surface. If an asymmetric particle changes shape or orientation during tracking, centroid will not always represent the same point in successive frames, and is not therefore a wholly valid measure (Uttenweiler et al., 2000). Actin filaments are only 7 nm wide, and can therefore be considered a long string of closely spaced point sources. Thus, if our data on centroid performance with point-source objects is applicable to actin filaments, then we can assume that the algorithm is both significantly biased, and on the ascending limb of its variance versus S/N relationship. The resulting bias error may explain recent observations of periodic motion of actin filaments (deBeer et al., 1997), which can make an actin filament appear to "jump" from pixel to pixel in the image. The leading and trailing ends of the filament would appear to move at the same overall rate, but with different phases depending on their positions relative to the imaging array. Only through simulations of these particular cases will we identify the best tracking method. However, our preliminary data on tracking actin filaments suggests that SAD may be the algorithm of choice among those tested in this study. A recent study has also shown excellent results using optical flow techniques (Uttenweiler et al., 2000).

We have extensively compared the efficacy of specific algorithms for determining the displacement of single particles. Our study is not a comprehensive comparison of the many different specific implementations of these algorithms, nor have we accounted for two other important aspects of particle tracking. First, we did not examine additional sources of noise and error. These might include noise resulting from analog or digital video encoding (such as interlace), spatially correlated noise common in ICCDs, and optical aberrations causing asymmetrical point spread functions. All these may differentially affect particle-tracking algorithms. Neither did we examine the plethora of techniques for image segmentation—identification and discrimination of many particles in a single set of images. Nonetheless, the algorithm that determines particle displacement defines the practical limit of particle tracking. The limiting bias and S/N values reported here should serve as a useful guide to the investigator both in choosing an algorithm and determining the performance limits and consequences of their existing software. We highly recommend the algorithm of direct Gaussian fit whenever a fluorescent point source is being tracked, particularly when S/N is limiting.

This work was supported by National Institutes of Health grants AR45604 and HL64381.

REFERENCES

Anderson, C. M., G. N. Georgiou, I. E. G. Morrison, G. V. W. Stevenson, and R. J. Cherry. 1992. Tracking of cell surface receptors by fluorescence digital imaging microscopy using a charge-coupled device camera. *J. Cell Sci.* 101:415–425.

Allersma, M. W., F. Gittes, M. J. deCastro, R. J. Stewart, and C. F. Schmidt. 1998. Two-dimensional tracking of ncd motility by back focal plane interferometry. *Biophys. J.* 75:1074–1085.

Bohs, L. N., B. H. Friemel, B. A. McDermott, and G. E. Trahey. 1993. A real time system for quantifying and displaying two-dimensional velocities using ultrasound. *Ultrasound Med. Biol.* 19:751–761.

deBeer, E. L., A. M. A. T. A. Sontrop, M. S. Z. Kellermayer, C. Galambos, and G. H. Pollack. 1997. Actin-filament motion in the in vitro motility assay has a periodic component. *Cell Motil. Cytoskel.* 38:341–350.

deJong, P. G. M., T. Arts, A. P. G. Hoeks, and R. S. Reneman. 1990. Determination of tissue motion velocity by correlation interpolation of pulsed ultrasonic echo signals. *Ultrason. Imaging.* 12:84–98.

Friemel, B. H., L. N. Bohs, and G. E. Trahey. 1995. Relative performance of two-dimensional speckle-tracking techniques: normalized correlation, non-normalized correlation and sum-absolute difference. *IEEE Ultrason. Symp.* 2:1481–1484.

Gelles, J., B. J. Schnapp, and M. P. Sheetz. 1988. Tracking kinesin-driven movements with nanometre-scale precision. *Nature.* 331:450–453.

Ghosh, R. N., and W. W. Webb. 1994. Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules. *Biophys. J.* 66:1301–1318.

Goulian, M., and S. M. Simon. 2000. Tracking single proteins within cells. *Biophys. J.* 79:2188–2198.

Guilford, W. H., and R. W. Gore. 1995. The mechanics of arteriole-interstitium interaction. *Microvas. Res.* 50:260–287.

Kubitscheck, U., O. Kückmann, T. Kues, and R. Peters. 2000. Imaging and tracking of single GFP molecules in solution. *Biophys. J.* 78:2170–2179.

Kusumi, A., Y. Sako, and M. Yamamoto. 1993. Confined later diffusion of membrane receptors as studied by single particle tracking (nanovid microscopy). Effects of calcium-induced differentiation in cultured endothelial cells. *Biophys. J.* 65:2021–2040.

Lee, G. M., A. Ishihara, and K. A. Jacobson. 1991. Direct observation of Brownian motion of lipids in a membrane. *Proc. Nat. Acad. Sci. U.S.A.* 88:6274–6278.

Papoulis, A. 1984. Probability, random variables, and stochastic processes. T. M. Slaughter, editor. McGraw-Hill, Inc., New York. 379–385.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1997. Numerical Recipes. Cambridge University Press, New York.

Ryan, T. A., P. J. Millard, and W. W. Webb. 1990. Imaging [Ca^{2+}] dynamics during signal transduction. *Cell Calcium.* 11:145–155.

Saxton, M. J., and K. Jacobson. 1997. Single particle tracking: applications to membrane dynamics. *Ann. Rev. Biophys. Biomol. Struc.* 26:373–399.

Schütz, G. J., H. Schindler, and T. Schmidt. 1997. Single-molecule microscopy on model membranes reveals anomalous diffusion. *Biophys. J.* 73:1073–1080.

Smith, M. J., E. L. Berg, and M. L. Lawrence. 1999. A direct comparison of selectin-mediated transient, adhesive events using high temporal resolution. *Biophys. J.* 77:3371–3383.

Sonnleitner, A., G. J. Schütz, and T. Schmidt. 1999. Free Brownian motion of individual lipid molecules in biomembranes. *Biophys. J.* 77: 2638–2642.

Uttenweiler, D., C. Veigel, R. Steubing, C. Götz, S. Mann, H. Haussecker, B. Jähne, R. H. A. Fink. 2000. Motion determination in actin filament fluorescence images with a spatio-temporal orientation analysis method. *Biophys. J.* 78:2709–2715.

Walker, W. F., and G. E. Trahey. 1994. A fundamental limit on the performance of correlation based phase correction and flow estimation techniques. *IEEE Trans. Ultrason. Ferroelec. Freq. Cont.* 41: 644–654.

Warshaw, D. M., E. Hayes, D. Gaffney, A. M. Lauzon, J. Wu, G. Kennedy, K. Trybus, S. Lowey, and C. Berger. 1998. Myosin conformational states determined by single fluorophore polarization. *Proc. Nat. Acad. Sci. U.S.A.* 95:8034–8039.

Work, S. S., and D. M. Warshaw. 1992. Computer-assisted tracking of actin filament motility. *Anal. Biochem.* 202:275–285.

Young, I. T. 1996. Quantitative microscopy. *IEEE Eng. Med. Bio.* 15: 59–66.