Available online at www.sciencedirect.com

ScienceDirect

Physica A 376 (2007) 117–132

PHYSICA A

www.elsevier.com/locate/physa

Discretization errors in particle tracking

G. Carmon, N. Mamman, M. Feingold*

Department of Physics and The Ilse Katz Center for Nanotechnology, Ben Gurion University, Beer Sheva 84105, Israel

Received 6 July 2006; received in revised form 8 September 2006
Available online 9 November 2006

Abstract

High precision video tracking of microscopic particles is limited by systematic and random errors. Systematic errors are partly due to the discretization process both in position and in intensity. We study the behavior of such errors in a simple tracking algorithm designed for the case of symmetric particles. This symmetry algorithm uses interpolation to estimate the value of the intensity at arbitrary points in the image plane. We show that the discretization error is composed of two parts: (1) the error due to the discretization of the intensity, $\mathbf{b}_D$ and (2) that due to interpolation, $\mathbf{b}_I$. While $\mathbf{b}_D$ behaves asymptotically like N^{-1} where N is the number of intensity gray levels, $\mathbf{b}_I$ is small when using cubic spline interpolation.

Keywords: Particle tracking; Microbeads; Discretization errors

1. Introduction

Microscopic particles are used in a wide range of experiments to monitor the dynamics of objects that cannot be directly visualized. Such particles are also used as tracers in flows [1] or as microprobes for various environments [2–4]. Quite often these particles are spherical beads of sizes ranging from 0.1 μm up to several μm. For example, attaching a bead to a single kinesin molecule Sheetz et al. [5] monitored its motion along a microtubule. For this purpose, they have developed a sub-pixel video tracking method that uses the correlation function between the images in the different frames. The precision was estimated to be of about 2 nm corresponding to about $\frac{1}{40}$ of the pixel size. Later work has revealed that kinesin advances in 8 nm steps [6].

An alternative approach relies on fluorescent particles or dyes. Tracking single fluorophores has allowed to monitor the dynamics of biomolecules in the cell membrane [7–10], in model membranes [11,12], in solution [13] and inside cells [14]. This field was reviewed in Refs. [15,16]. For example, in a recent study, tracking of single fluorophores was used to monitor the motion of myosin V on actin. Single rhodamine or Cy3 molecules were used to label one of the "walking legs" of myosin V [17]. Special care was required in order to prevent the fluorophore from bleaching and to optimize the signal-to-noise ratio (S/N). The tracking was done using a Gaussian fit to the intensity profile leading to a precision of 1.5 nm.

*Corresponding author. Tel.: +972 8 6461587; fax: +972 8 6472904.
E-mail address: mario@bgu.ac.il (M. Feingold).

0378-4371/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2006.10.048

The use of single fluorophores in biophysics is among the most modern techniques in the field and is pushing the limits of light detection technology. It is mainly limited by the low number of photons that the fluorophore emits before bleaching. This makes the photon counting noise the dominant source of error in single fluorophore tracking. On the other hand, in most optical and magnetic tweezing experiments, microbeads are used as handles through which external forces can be applied. Moreover, colloidal suspensions of microbeads are studied in order to understand the two-body and many-body interactions between the beads. Such suspensions were shown to display phase transitions between liquid-like and crystal-like configurations [18]. Video tracking of microbeads is required in order to characterize the system. In such experiments, imaging is typically achieved using brightfield microscopy where the photon counting noise is relatively low. Here, the other types of error will contribute and, in particular, the discretization error will play an important role in limiting the precision of tracking.

Quite a few tracking algorithms are used in experiments and their accuracy is not always well defined. The performance of four such algorithms was compared by Cheezum et al. [19]. They compared the cross-correlation [5,20,21], sum-absolute difference (SAD) [22], centroid [10,23] and direct Gaussian fit [12,24] algorithms and have shown the way in which the various sources of error contribute to the accuracy of the different techniques. The study of Ref. [19] was focused on finding the best way to monitor the dynamics of a single fluorophore. It was found that, due to the small S/N, the Gaussian fit is the most accurate algorithm for such experiments. For the case of microscopic particles that are larger than the point-spread function (PSF), they found that the cross-correlation method is the most accurate.

In video tracking there are several sources of error. It is customary to divide the errors in two categories: random and systematic. Random errors (noise) are referred to as limiting the *precision* of the method while systematic ones (bias) affect the *accuracy*. Although noise can be reduced by averaging, this limits the time resolution. Even in experiments where this is not an issue, averaging is limited by the stability of the system. On the other hand, one expects that a better understanding of the experimental system would allow to eliminate or at least reduce the bias. While each experimental system has its own characteristics, there is one source of bias that is generic, namely, the bias due to the digitization of the image, $\mathbf{b}$. This source of error is often ignored in tracking experiments. The purpose of this paper is to describe the behavior of the discretization bias and to show how it can be significantly reduced. For this we choose to use a simple algorithm that allows to track symmetric particles with sub-pixel accuracy, the symmetry algorithm. While its simplicity allows us to perform a detailed analysis, its general features may also apply to some of the other algorithms, in particular, to the SAD algorithm [22]. Moreover, as will be shown in Section 5, the error of the symmetry algorithm for microbeads is comparable to that of the cross-correlation algorithm that is currently considered to be the most precise. We show that for our algorithm the discretization bias, $\mathbf{b}$, is composed of two parts that to a good approximation are additive, namely, the bias due to the intensity discretization itself, $\mathbf{b}_D$, and that due to interpolation, $\mathbf{b}_I$. Asymptotically, $\mathbf{b}_D \propto N^{-1}$, where N is the number of gray levels. Moreover, in the case of linear interpolation, $\mathbf{b}_I \to N_E^2 g(\varepsilon)$ for $N_E \gg 1$, where N_E is the number of extrema of the intensity function and $g(\varepsilon)$ is an antisymmetric function that is almost entirely independent on the intensity function. The paper is organized as follows. In the next section we introduce the symmetry algorithm. Then, in Section 3 we analyze the behavior of the discretization bias for 1D images. Next, in Section 4 we show that many of the features of the 1D behavior are preserved in the 2D case. A brief experimental comparison of the different tracking errors is presented in Section 5. Section 6 contains our conclusions.

2. The symmetry algorithm

In the case of both microscopic beads that are larger than the PSF and point sources the ideal image has circular symmetry. This symmetry can be used to find the center of the particle using the axes of symmetry in both the x and y directions, $L_x\colon y = y_0$ and $L_y\colon x = x_0$. The center of the particle lies at the intersection of the two symmetry axes, (x_0, y_0) (see Fig. 1). Each of the symmetry axes is found by searching for the minimum of the symmetry functions, $S_y(\varepsilon_x)$ and $S_x(\varepsilon_y)$, where ε_x and ε_y generate sub-pixel shifts of the symmetry axes in the x and y direction, respectively. For the case of L_y,

$$S_y(\varepsilon_x) = \sum_{i=1}^{N_1} \sum_{j=-N_1}^{N_1} |f(i - \varepsilon_x, j) - f(-i - \varepsilon_x, j)|, \tag{1}$$

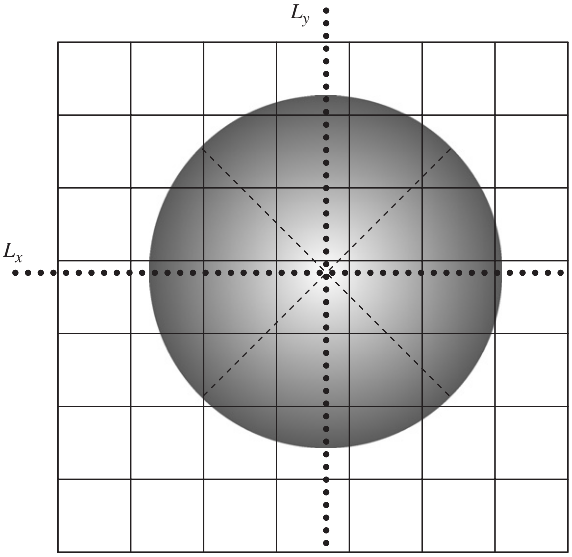

Fig. 1. Schematic description of the symmetry algorithm. For example, one shifts the L_y symmetry axis until the intensities on its two sides are most similar in the sense of Eq. (1).

where i runs over the pixels to the right of L_y, excluding the $i = 0$ pixels, namely, the pixels through which L_y passes, j runs along the columns of the entire image, $f(i - \varepsilon_x, j)$ is the intensity value at $(i - \varepsilon_x, j)$ and $2N_1 + 1$ is the size of the image in pixels. Since digital images only provide intensities of entire pixels, we obtain the value of $f(i - \varepsilon_x, j)$ by interpolation between the values of adjacent pixels. In this work, two types of interpolation were used: linear and natural cubic spline [25]. While the case of linear interpolation is more analytically tractable, the cubic spline interpolation leads to very small values of the corresponding bias, $\mathbf{b}_I$. S_x is obtained by exchanging the roles of i and j and of x and y in Eq. (1).

For a perfectly symmetric intensity function whose values are known at all (x, y) points (no pixels), the minimum of S_x and S_y corresponds to the exact center. However, in actual images the symmetry is weakly perturbed due to optical aberrations, noise and discretization effects. Assuming an ideal optical system that preserves the circular symmetry and ignoring the effect of experimental noise, we are left with the bias that is due to discretization, $\mathbf{b}$. It is defined as

$$\mathbf{b} = \boldsymbol{\varepsilon}_1 - \boldsymbol{\varepsilon}_0, \tag{2}$$

where $\boldsymbol{\varepsilon}_1$ is the measured shift obtained from the minimization of the experimentally measured, discretized S's and $\boldsymbol{\varepsilon}_0$ gives the position of the true symmetry center of the particle. In experiment, it is difficult to separate the different sources of error in order to understand the behavior of each one. However, this can be done in a model system and therefore, in the next section, we analyze the behavior of the bias for various types of intensity functions. We found that the bias behaves similarly in one and in two-dimensions and that the 2D case does not present any new features with respect to one-dimension.

3. The bias for the symmetry algorithm in 1D—theory and simulations

In what follows, we analyze the bias due to discretization for 1D intensity functions. In particular, we will illustrate the behavior of the bias using three different symmetric functions, f_1, f_2, and f_3, where the first is a Gaussian, the second is an exponent that decays in an oscillatory manner and the third is a cosine,

$$f_1(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \tag{3}$$

$$f_2(x) = \frac{1}{2}\exp\left(-\frac{|x|}{a}\right)\cos\left[\frac{\pi x(n+1/2)}{A}\right] + \frac{1}{2}, \tag{4}$$

$$f_3(x) = \frac{1}{2}\cos\left[\frac{\pi x(n+c/2)}{A}\right] + \frac{1}{2}. \tag{5}$$

While the Gaussian represents a good approximation to the PSF, f_2 and f_3 are reminiscent of the images of defocused beads that display multiple interference fringes (see Fig. 12b). We have studied a variety of functions and found that f_1, f_2 and f_3 are representative of certain function families: (1) monotonically decaying on each side of the symmetry center, (2) decaying in an oscillatory manner and (3) oscillating at constant amplitude.

We apply to these functions a discretization similar to that of a noise-less 1D digital camera with N gray levels and $2N_1+1$ pixels. In a digital CCD, the value of the intensity function in a given pixel results from integrating the light intensity over the active area of the pixel. We approximate this process by taking the value of the intensity function at the center of the pixel to represent the entire pixel. The pixels are chosen to be of unit length and centered on integers, such that, in the first digitization step, D_1,

$$D_1{:}f_l(x) \to f_l(i), \tag{6}$$

where $l = 1,2,3$ and $-N_1 \leqslant i \leqslant N_1$. The second digitization step, D_2, approximates the intensity of the pixel by the closest available gray level,

$$D_2{:}f_l(i) \to \tilde{f}_l(i), \tag{7}$$

where $\tilde{f}_l(i) = N^{-1}\mathrm{fix}[Nf_l(i)]$ and the function fix rounds the value of its argument to the nearest integer value. Since the functions we study are symmetric and their symmetry center is located at the middle of the central pixel, $i = 0$, the symmetry is preserved under discretization, D, where $D = D_2 \cdot D_1$. However, such coincidence between the symmetry center of an image and the center of some pixel center has zero probability when imaging a symmetric object. In order to include in the model the difference between the symmetry center of the image and the center of the pixel, we shift the image function by $\varepsilon_0 \leqslant 1$ from the origin, $T_{\varepsilon_0}{:}f(x) \to f(x-\varepsilon_0)$ and only then discretize using the operator $D, \tilde{f}_l(i-\varepsilon_0) = D \cdot T_{\varepsilon_0} f_l(x)$. Due to the mismatch between the pixel array and the symmetry center, the discretization will perturb the symmetry of the function such that $\tilde{f}_l(i-\varepsilon_0)$ is only approximately symmetric. As a consequence, searching for ε_0 using the symmetry algorithm of Section 2 will lead to a slightly different value, ε_1. The bias is defined as in Eq. (2), $b = \varepsilon_1 - \varepsilon_0$. In what follows, we study the behavior of the bias as the various parameters of the problem change, namely, the number of gray levels, N, the number of pixels, N_1 and the various parameters of the intensity functions, $f_l(x)$.

3.1. Dependence of the bias, b, on the number of gray levels, N

Throughout the rest of the paper, we express all lengths, e.g. ε_0, ε_1 and b, in units of pixels. In Fig. 2 we show the behavior of b for f_1 and f_2 as a function of N for a particular shift, $\varepsilon_0 = 0.1$, and linear interpolation. For both functions the bias oscillates wildly. However, while for the Gaussian it seems to oscillate around zero, in the case of f_2 it clearly has a finite average value. In fact, the average of b in Fig. 2a is $\langle b_{1,L}\rangle = 1.81 \times 10^{-4}$ and in Fig. 2b it is $\langle b_{2,L}\rangle = 4.66 \times 10^{-3}$, where the first index of $\langle b\rangle$ indicates to which function it corresponds and the second which type of interpolation was used ($L =$ linear, $S =$ natural cubic spline). When the same calculation is made with spline interpolation the oscillations appear to be quite similar in size but the corresponding averages are different, namely, $\langle b_{1,S}\rangle = -1.97 \times 10^{-5}$ and $\langle b_{2,S}\rangle - 2.08 \times 10^{-4}$. This suggests that the fluctuations in the bias are determined by the D_2 operator that transforms the function into its approximation in terms of N gray levels, while the average bias is due to interpolation. To verify this assumption, we can, in our computation, turn off the D_2 step to obtain a bias that is due to interpolation alone, b_I. Indeed, for the case of the linear interpolation the values of b_I that we obtain are similar to those of the corresponding $\langle b\rangle$. Specifically, $b_{I,1,L} = 1.87 \times 10^{-4}$ and $b_{I,2,L} = 4.00 \times 10^{-3}$. However, for the case of the spline interpolations, the values of b_I that we find are completely different than those of $\langle b\rangle$, that is, $b_{I,1,S} = 1.36 \times 10^{-7}$ and $b_{I,2,S} = -1.18 \times 10^{-6}$. The discrepancies between $\langle b\rangle$ and b_I are due to the finite size effects in

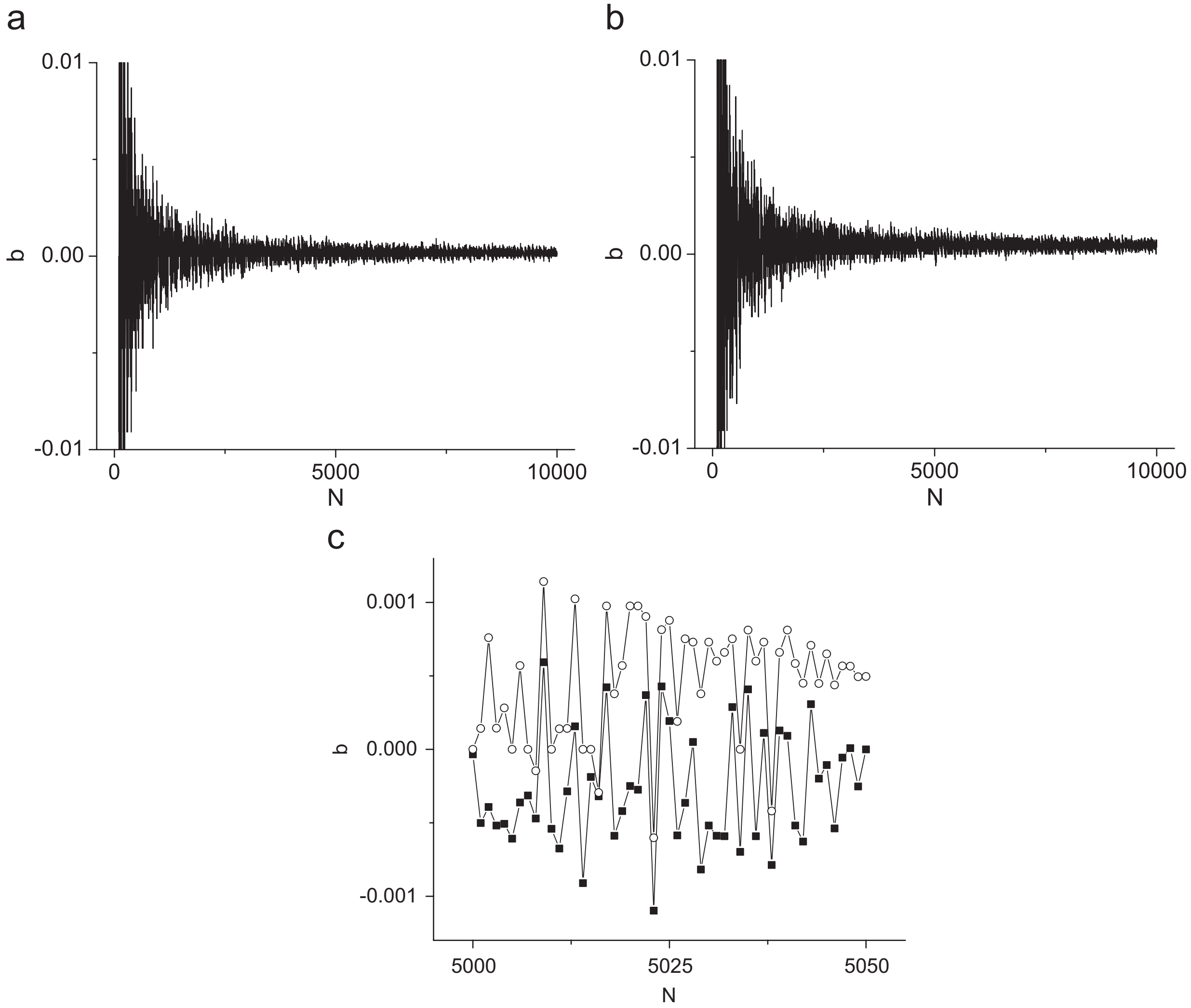

Fig. 2. Variation of the bias as a function of the number of gray levels, N, using linear interpolation for $N_1 = 40$, $\varepsilon_0 = 0.1$ and (a) Gaussian, f_1, with $\sigma = 10$, (b) decaying cosine, f_2, with $a = 10$, $A = 40$ and $n = 2$. At small N, in both (a) and (b), the bias exceeds the range of the y-axis that is shown. (c) Same as in (b) only that here the range of N values is small (line with circles). The second curve corresponds to the case of spline interpolation for f_2 (line with full squares). Notice that the two lines are displaced with respect to each other due to the different average values of the bias for linear and spline interpolation, $b_{I,2,L} \neq b_{I,2,S}$. The bias is expressed in units of pixels.

the averaging of $\langle b \rangle$. These are relatively small whenever b_I is large as in the case of linear interpolation, but become significant for the spline case. Performing the average of b at large values of N we obtain values that are closer to those of b_I. In particular, when b is averaged over $10^7 \leqslant N \leqslant 1.0001 \times 10^7$ we obtain $\langle b_{1,L} \rangle = 1.87 \times 10^{-4}$, $\langle b_{2,L} \rangle = 4.01 \times 10^{-3}$, $\langle b_{1,S} \rangle = 1.87 \times 10^{-7}$ and $\langle b_{2,S} \rangle = -1.36 \times 10^{-6}$ showing clear convergence towards the corresponding b_I's. This indicates that the average bias is due to the interpolation error while the fluctuations around this average are due to the discretization error and that one can approximately decompose b, such that,

$$b \cong b_I + b_D, \tag{8}$$

where b_D is the bias that is due to the D_2 operator. In our simulation of the symmetry algorithm the interpolation is performed after the discretization step and therefore its result depends on the discretization. This leads to some coupling between the b_I and b_D and to the approximate equality sign in

Eq. (8). The fluctuations in the bias for the linear and spline interpolations are compared in Fig. 2c for the case of the decaying cosine function, f_2. We notice that the magnitude of b_D seems to behave similarly for both types of interpolation (see also Fig. 4). Moreover, in the case of the Gaussian, the linear and spline interpolations lead to fluctuations that are practically identical. As will be shown in the next section, for the case of linear interpolation, Eq. (8) is due to an expansion of S. We find that b_I vanishes at the lowest order of this expansion.

3.2. *The structure of the symmetry function, S, for linear interpolation*

In order to understand the behavior of the bias, it is important to first describe the structure of the symmetry function and in particular, the way its minimum is determined. For simplicity, this analysis is restricted to the case of the linear interpolation. For a given initial shift, ε_0, the symmetry function, S, can be viewed as depending on the value of the restoring back-shift, ε_1. Before any errors are introduced by discretization

$$S(\varepsilon_1) = \sum_{i=1}^{N_1} |f(i-\varepsilon_0+\varepsilon_1) - f(-i-\varepsilon_0+\varepsilon_1)|. \tag{9}$$

However, the value of the function at $x = \pm i - \varepsilon_0 + \varepsilon_1$ is unknown when the pixel centers are at $x = \pm i - \varepsilon_0$ and is approximated in the symmetry algorithm by linear interpolation, such that

$$S(\varepsilon_1) \simeq \sum_{i=1}^{N_1} |f(i-\varepsilon_0) + \varepsilon_1[f(i+1-\varepsilon_0) - f(i-\varepsilon_0)] - f(-i-\varepsilon_0) - \varepsilon_1[f(-i+1-\varepsilon_0) - f(-i-\varepsilon_0)]|. \tag{10}$$

The next error is due to the discretization of the intensities, f. We denote these errors as Δf_i. To estimate the expression of Eq. (10) we assume that the products of Δf_i and ε_1 are of higher order and therefore can be neglected. Moreover, we expand the f's to first order in ε_0 and $1-\varepsilon_0$ to obtain

$$S(\varepsilon_1) \simeq \sum_{i=1}^{N_1} |\Delta f_i - \Delta f_{-i} - 2(\varepsilon_0 - \varepsilon_1) f'(i)|, \tag{11}$$

where $f'(i)$ is the derivative and we used the symmetry of f. Each of the terms in the sum over pixels is linear in b decreasing with slope $2f'(i)$ until it vanishes at $\varepsilon_{1,i}$ where it changes slope to $-2f'(i)$. Moreover

$$b_i \equiv \varepsilon_{1,i} - \varepsilon_0 = \frac{\Delta f_{-i} - \Delta f_i}{2f'(i)}. \tag{12}$$

To estimate the b_i's, we note that the values of an arbitrary function are unrelated to the gray levels imposed by the camera. It is therefore a reasonable assumption that, on average, these values are homogeneously distributed between each two adjacent gray levels, $f_{\pm i-\varepsilon_0} \in (k_\pm - \frac{1}{2N}, k_\pm + \frac{1}{2N})$, $\tilde{f}(\pm i - \varepsilon_0) = k_\pm$ and $\Delta f_\pm \in (-\frac{1}{2N}, \frac{1}{2N})$. One expects that this approximation becomes more accurate as N increases. Moreover, for large enough ε_0 and N, $k_+ \neq k_-$, leading to uncorrelated Δf_i and Δf_{-i}. In this regime, averaging over consecutive values of N, e.g. in Fig. 2, gives

$$\langle b_i \rangle = 0, \tag{13}$$

$$Sd(b_i) = \frac{1}{2\sqrt{6} N f'(i)}, \tag{14}$$

where Sd denotes the standard deviation.

Next step is to understand the way that the individual terms in Eq. (11) combine to determine the behavior of $S(\varepsilon_1)$. As can be seen in Fig. 3, $S(\varepsilon_1)$ is composed of straight segments that start with a large negative slope that discontinuously changes and progressively becomes smaller. At the minimum, the slope changes sign and afterwards grows towards a large positive value. The discontinuous changes in the slope of $S(\varepsilon_1)$ occur at $\varepsilon_{1,i}$. There, its slope changes by $4f'(i)$ (see Eq. (11)). In particular, one of the pixels is responsible for changing the sign of the slope at the minimum. We refer to it as the dominant pixel, $i = i_d$, because it determines the value of the overall bias, $b = b_{i_d}$. To find the dominant pixel, we need to understand both the sequence of the b_i's

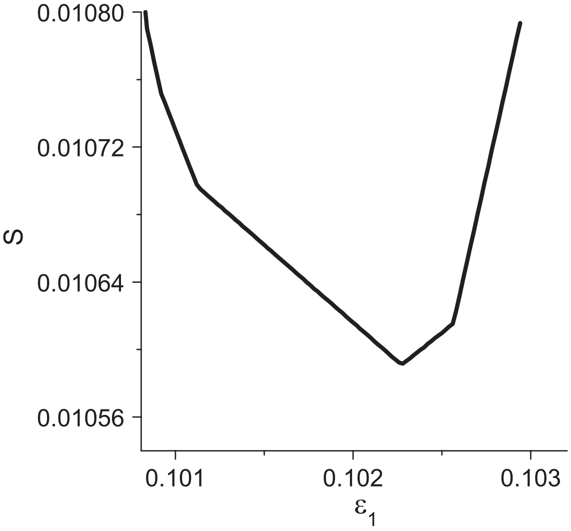

Fig. 3. The symmetry function for the Gaussian, f_1, in the vicinity of its minimum. The parameters are the same as in Fig. 2a only that here N is fixed, $N = 1000$.

when arranged in order of increasing value and the corresponding way in which the slope of $S(\varepsilon_1)$ decreases. In order to illustrate these aspects we use the case of the Gaussian, f_1, as an example.

For the Gaussian, the derivative has a single extremum for $x>0$ at $x=\sigma$. For a given N, on average half of the b_i's are positive and half are negative. On the other hand, the slope of $S(\varepsilon_1)$, $A(b)$, starts from $2\sum_{i=1}^{N_1} f'(i)$ at $\varepsilon_1 \ll \varepsilon_0$ and then increases with ε_1 by $4f'(j)$ at each point where $b = b_j$. Therefore, in order to reach a vanishing slope we require that

$$A(b) = 2\sum_{i=1}^{N_1} f'(i) - 4\sum_{j'=1}^{\sigma-K_1} f'(j') - 4\sum_{j'=\sigma+K_2}^{N_1} f'(j') = 0, \tag{15}$$

where j' runs over the b_i's in order of increasing value and $K_{1,2}$ define the number of b_i's required to balance the first term. The second and third sums of Eq. (15) contain much less terms than the first one. The index j' scans all the pixels in the ranges $(1, \sigma - K_1)$ and $(\sigma + K_2, N_1)$ but picks only the b_i's with the largest absolute value and negative sign. On average there are $(N_1 - K_1 - K_2)/2$ terms in the two j' sums. Notice that the arguments of the slope, $A(b)$, and that of the symmetry function itself, $S(\varepsilon_1)$, are equivalent because here ε_0 is a constant and $b = \varepsilon_1 - \varepsilon_0$. Since on average there are $N_1/2$ negative b_i's, Eq. (15) is on average satisfied whenever $K_1 = K_2 = 0$. In other words, the dominant pixel is the one for which $f'(i)$ is maximal, $i_d \simeq i_\sigma \equiv \mathrm{fix}(\sigma)$. Using Eq. (14) for f_1, we obtain

$$Sd(b) = Sd(b_D) \approx \frac{1}{2}\sqrt{\frac{e}{6}\frac{\sigma}{N}} \simeq 0.34\frac{\sigma}{N}, \tag{16}$$

where we follow the definitions of Section 3.1 and identify the fluctuating part of b as being due to discretization, b_D. The $\approx$ sign of Eq. (16) indicates that this is only an estimate for $Sd(b_D)$ since three factors that contribute to determining its actual value have not been included in the derivation.

First, we have ignored the fact that statistical fluctuations in the discretization errors, Δf_i, will also lead to fluctuations in the position of the dominant pixel, i_d. The pixels in the neighborhood of i_σ will compete with each other for the role of the dominant pixel (the one with the smallest b_i). This competition is manifested as fluctuations in the ordering of the b_i's (see Eq. (12)). It will reduce the value of the bias. While the correction to the bias will not affect its dependence on N, it can lead to a different dependence on σ because more pixels will participate in the competition the broader the Gaussian. For the same reason, $Sd(b_D)$ will also depend on N_1.

A second effect that has been neglected in the derivation of Eq. (16) consists of fluctuations in the number of negative b_i's, $N_{1,-}$. If $N_{1,-} \neq N_1/2$, the dominant pixel will not be the one with maximal $f'(i)$, $i_d \neq i_\sigma$ even if all

$|\Delta f_{-i} - \Delta f_i|$ are equal, $|\Delta f_{-i} - \Delta f_i| = \eta$. Instead, i_d will change such as to correspond to the range where $S(\varepsilon_1)$ has a vanishing slope (see Eq. (15)). While this effect does not change $\langle b_D \rangle$, it leads to a variability in the dominant pixel, $i_d = i_\sigma \pm \Delta i$, which, in turn, leads to variability in the derivative, $f'(i_d)$. Now both $\Delta f_{\pm i}$ and $f'(i_d)$ need to be regarded as random variables and this will lead to a larger $Sd(b_D)$. However, since $f'(i_d)$ is fluctuating around its maximum its contribution to $Sd(b_D)$, Δb_2, vanishes to first order. To second order,

$$\Delta b_2 = \frac{1}{8}\sqrt{\frac{e}{6}\frac{N_1}{\sigma}}\frac{1}{N}. \tag{17}$$

Finally, a third contribution to b_D that was ignored in Eq. (16) is the presence of small intensity values in the pixels that are far from the origin in the case of a strongly decaying function. If N is not large enough, such values will be quantized to zero. On the other hand, as N grows these pixels will eventually start to contribute to the symmetry function in a non-trivial way, leading to an effective increase in N_1, $N_1(N)$. This effect will lead to deviations from the N^{-1} dependence of $Sd(b_D)$ for the Gaussian at small σ, $\sigma \ll N_1$ (see Fig. 8a).

It is important to note that Eq. (13) apparently contradicts our numerical observation of Section 3.1 where we found that $\langle b \rangle = b_I \neq 0$. This will be resolved by going to the next order in the expansion used in Eq. (11) which gives a non-vanishing contribution to b_I (see Section 3.3). Another important observation, is that the only explicit manifestation of the Gaussian functional form in the derivation of Eq. (16) is the maximal value of the derivative. Therefore, within the approximation described above, this provides an estimate for $Sd(b_D)$ for an arbitrary function when replacing the corresponding maximal derivative

$$Sd(b_D) \approx \frac{1}{2\sqrt{6}N Max(|f'|)}. \tag{18}$$

In Fig. 4 we test the validity of Eq. (18) for the case of f_1 and f_2. Here the value of b_D is obtained from Eq. (8) where b_I is computed by turning off discretization and the approximate equality is assumed to be exact. The standard deviation of b_D is computed by averaging over consecutive intervals of N of 10^2. While for both f_1 and f_2 the N^{-1} behavior is clear, the accuracy of our estimate is much better for the decaying exponent, f_2 than for the Gaussian, f_1.

We can further extend the applicability of Eq. (18) to include the effect of random noise, δf_i, that has the same standard deviation in all pixels (see Section 5). Such random noise will simply add to the one due to discretization and the standard deviation of the sum, $\sqrt{[Sd(\Delta f_{-i} - \Delta f_i)]^2 + [Sd(\delta f_{-i} - \delta f_i)]^2}$, should replace the factor $(2\sqrt{3}N)^{-1}$ in Eq. (18) to account for the combined effects on the fluctuating part of the error. Note that, while δf_i is intrinsically random, Δf_i is not and only depends in a complex way on the parameters of the problem.

3.3. The interpolation bias, b_I

For practical purposes, the most important fact concerning the interpolation bias, b_I, is that when using cubic spline interpolation it becomes negligible (see Section 3.1). For example, for the case of f_2 with the parameters of Fig. 2b, only CCD's with more than 19 bit/pixel would reach the range where $b_{I,2,S} \sim b_D$. Moreover, in most applications such low levels of discretization errors are most likely negligible relative to the other sources of error, e.g. noise. On the other hand, since spline interpolation is computationally expensive it is worthwhile to understand the behavior of b_I for the case of the linear interpolation.

As already pointed out in Section 3.2, Eq. (13) for $i = i_d$ contradicts our findings of Section 3.1 where we showed that $\langle b \rangle \neq 0$. There we also showed evidence that the non-vanishing of the average bias is due to the interpolation. Therefore, in order to resolve this contradiction, we return to Eq. (10) and, in the derivation leading to Eq. (11), we turn off the effect of intensity discretization, $\Delta f_i = 0$. In this case, to first and second order in ε_0 and $1 - \varepsilon_0$ the expansion of Eq. (10) leads to a vanishing bias. We therefore pursue the expansion up to third order to obtain that the ith term in the sum of Eq. (10) vanishes when

$$b_i \simeq -\varepsilon_0 \frac{f'''(i)(1 - 3\varepsilon_0 + 2\varepsilon_0^2)}{6f'(i) + f'''(i)(1 - 3\varepsilon_0 + 3\varepsilon_0^2)}. \tag{19}$$

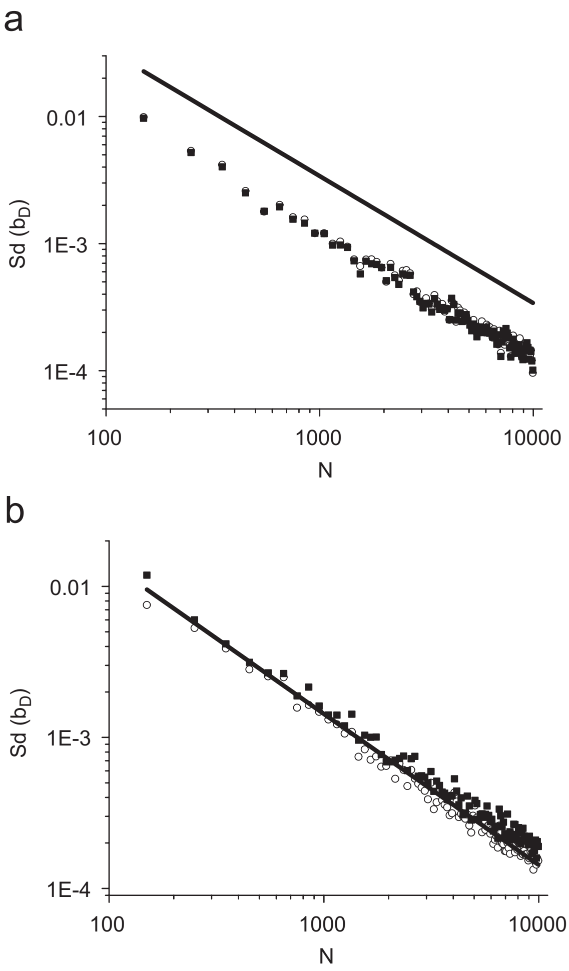

Fig. 4. Standard deviation of b_D (in pixels) as a function of the number of gray levels, N, using either linear (open circles) or cubic spline (full squares) interpolation. The parameters are the same as in Figs. 2a and b, respectively. (a) Gaussian, f_1, with $\sigma = 10$, (b) decaying cosine, f_2, with $a = 10$, $A = 40$ and $n = 2$. The straight line represents the prediction of Eq. (18).

As in Section 3.2, in what follows we illustrate the behavior of Eq. (19) for the case of the Gaussian, f_1. In this case, the b_i's of Eq. (19) are ordered from the outermost pixel towards the origin, that is, $i > j \Rightarrow b_i > b_j$. Moreover, the structure of the symmetry function, $S(\varepsilon_1)$, is qualitatively similar to that of Fig. 3 and here, as well, there will be a dominant pixel, $i = i_d$, which determines the value of the total bias, b_I. Using a similar approach to that of Eq. (15), we need to find the pixel which leads to a vanishing slope for $S(\varepsilon_1)$. Since here the b_i's are ordered,

$$A(b) = 2\sum_{i=1}^{N_1} f'(i) - 4\sum_{i=i_d}^{N_1} f'(i) = 0. \tag{20}$$

In the limit $N_1 \gg \sigma$ and replacing the sums with integrals we obtain

$$2f(i_d) \simeq 1, \tag{21}$$

where we used the normalization $f(0) = 1$ (same as in Eqs. (3)–(5)). This leads to

$$i_d \simeq \text{fix}(\sqrt{\ln 4}\sigma), \tag{22}$$

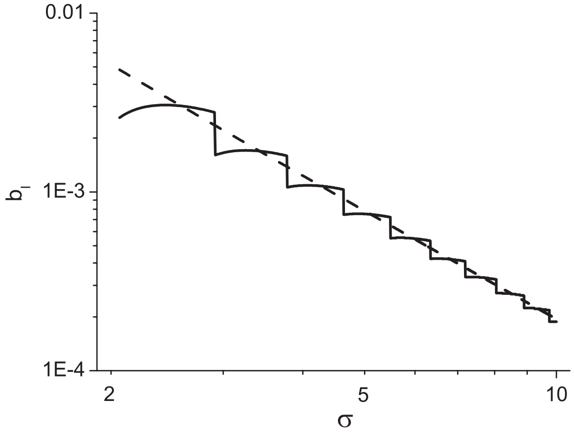

Fig. 5. Variation of b_I (in pixels) as a function of a parameter of the intensity function for the case of the Gaussian, f_1 (full line). We change its width, σ, using linear interpolation, $\varepsilon_0 = 0.1$ and $N_1 = 40$. The prediction of Eq. (23) is also shown (dashed line).

and correspondingly to

$$b_I \simeq \frac{\alpha(\varepsilon_0)}{3.71\sigma^2 - \beta(\varepsilon_0)}, \tag{23}$$

where $\alpha(\varepsilon_0) = \varepsilon_0(1 - 3\varepsilon_0 + 2\varepsilon_0^2)$ and $\beta(\varepsilon_0) = 1 - 2\varepsilon_0 + 3\varepsilon_0^2$. For $\sigma > 1$, β is small relative to the first term in the denominator of Eq. (23) and b_I tends to a power law dependence on σ, $b_I \propto \sigma^{-2}$. Although the prediction of Eq. (23) is in agreement with the corresponding numerical results (see Fig. 5), we find an additional pattern of equidistant discontinuities in the value of b_I. These occur at the values of σ where the dominant pixel changes. Here, $S(b_{i_d}) = S(b_{i_d+1})$ and its minimum degenerates into a segment with vanishing slope. These discontinuity points are the solutions of Eq. (20). Inverting Eq. (22), we obtain that the jumps are expected at fixed intervals in σ, $\Delta\sigma \simeq 0.85$. Indeed, numerically we find that the discontinuities in b_I occur at values of σ, σ_i, where $\Delta\sigma_i \equiv \sigma_{i+1} - \sigma_i \to 0.85$ as σ grows ($\Delta\sigma_3(\sigma_3 = 2.91) = 0.864$, but $\Delta\sigma_6(\sigma_6 = 5.49) = 0.854$). The variation of b_I is much larger at σ_i than in the range between discontinuities, leading to a staircase-like behavior.

For other functions, e.g. f_2, the b_i's of Eq. (19) are not monotonic functions of i. As a result, the pattern of discontinuities is more complex than that of the Gaussian. Nevertheless, for each function, the behavior of b_I as a parameter of the function varies can be deduced. We found no indication of generic behavior. However, in the next section, we show that the dependence of b_I on ε_0 and N_E is largely independent on the functional form of the intensity.

3.4. Dependence of the interpolation bias, b_I, on parameters

Interpolation, and in particular, linear interpolation is most inaccurate in the neighborhood of extrema. If consecutive pixels are located on opposite sides of an extremum of the intensity function, linear interpolation will place the approximated value on the straight line that joins the values in the pixels, thereby truncating the extremum. It is therefore natural to expect that the interpolation bias, b_I, will be dominated by the contribution from the extrema and that $b_I \propto g_E(N_E)$ where N_E is the number of extrema of f and g_E is a function that remains to be determined. In Fig. 6 we show the behavior of b_I for the decaying exponential, f_2, and two different cosine functions, f_3, at $\varepsilon_0 = 0.1$. For large N_E the function g_E approaches a power law

$$g_E(N_E) \to CN_E^{\alpha} \quad \text{for } N_E \gg 1, \tag{24}$$

where for the cosine with $A = 40$ the best fit gives $\alpha = 1.996 \pm 0.011$ and $C = (1.922 \pm 0.005) \times 10^{-4}$. This result suggests that asymptotically $\alpha = 2$. As can be seen in Fig. 6, even for small number of extrema, $N_E < 10$, b_I is nearly in its asymptotic regime.

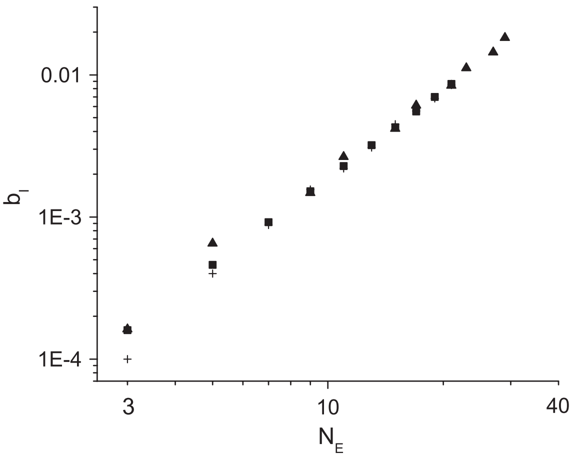

Fig. 6. Variation of b_I (in pixels) as a function of the number of extrema, N_E, using linear interpolation, $N_1 = 40$, $\varepsilon_0 = 0.1$ and a decaying cosine, f_2, with $a = 10$ and $A = 40$ (pulses), a cosine, f_3, with $A = 40$ and $c = 1$ (squares) and a second cosine with $A = 27$ and $c = 0$ (triangles).

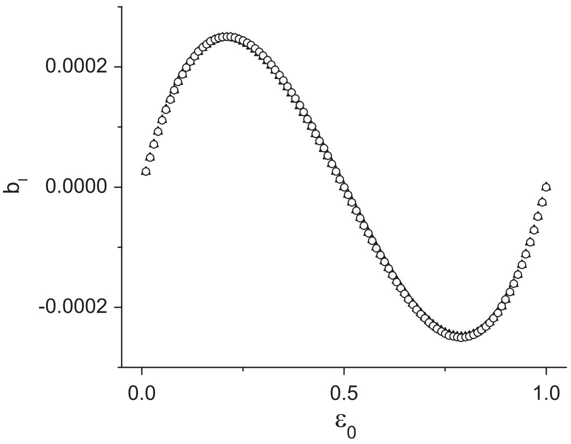

Fig. 7. Variation of b_I (in pixels) as a function of ε_0 using linear interpolation, $N_1 = 40$ and a Gaussian, f_1, with $\sigma = 10$ (circles), a decaying cosine, f_2, with $a = 10$, $A = 40$ and $n = 2$ (pulses) and a cosine, f_3, with $A = 40$, $c = 1$ and $n = 5$ (triangles). The values of b_I for f_2 and f_3 have been normalized such that they are equal to those of the Gaussian at $\varepsilon_0 = 0.1$. Notice that for f_2, $N_E = 5$ and for f_3, $N_E = 11$. Circles almost precisely cover the other symbols. Differences between the data points at fixed ε_0 are nowhere larger than 1.5×10^{-6}.

In addition to the dependence on N_E, b_I also depends on the value of ε_0. In Fig. 7 we show that $b_I \propto g(\varepsilon_0)$, where $g(\varepsilon_0)$ is weakly dependent on the form of the intensity function. From the definition of the symmetry function, Eq. (1), it can be shown that $g(\varepsilon_0)$ is antisymmetric, $g(\varepsilon_0) = -g(-\varepsilon_0)$.

4. The bias for the symmetry algorithm in 2D—theory and simulations

In two dimensions, the behavior of $\mathbf{b} = (b_x, b_y)$ is similar to the 1D case discussed in the previous section if each of the components is viewed separately. E.g. for b_x we obtain similar equations for the symmetry function as Eqs. (9)–(11). The only difference is the additional summation along the y direction, $\sum_{j=-N_1}^{N_1}$, and the fact that the derivatives of f become here partial derivatives with respect to x, $f' \to f_x$. This also applies for Eqs. (12), (18) and (19). The difference between 1D and 2D will appear when determining the dominant pixel, (i_d, j_d). In 2D, there are $2N_1^2$ single pixel biases, $b_{x,D}(i,j)$'s and $b_{x,I}(i,j)$'s, that have to be arranged in order of increasing value to find which one determines the minimum of the symmetry function. As in 1D, the behavior of the bias is particularly simple for the axially symmetric 2D Gaussian. Here as well, the discretization bias, $b_{x,D}$, is approximated by the contribution from the pixel at $(\mathrm{fix}(\sigma), 0)$ that has the largest derivative. As a result, the estimate for $Sd(b_{x,D})(N)$ is the same as in 1D and the exact $Sd(b_{x,D})(N)$ behaves similarly to that in 1D (see Fig. 8a). However, in 2D there is some additional structure beyond the N^{-1} behavior which is due to the low intensity pixels effect (see Section 3.2). Moreover, in 2D, $Sd(b_{x,D})(N)$ is smaller than in 1D due to the large increase in the number of pixels that compete to be dominant (see Section 3.2). As for the $b_{x,I}(i,j)$'s, they are independent of j and depend on i in the same way as in 1D. Therefore, the interpolation bias for the Gaussian, $b_{x,I}$, has instead of a single dominating pixel an entire column of such pixels, (i_d,j), where i_d is given by Eq. (22). Together these determine the minimum of the symmetry function. This column plays the same role in 2D as the single determining pixel, i_d, in 1D. It is therefore to be expected that there will be no difference between the 1D and 2D behaviors of $b_{x,I}(\sigma)$ (see Fig. 9). The identical behavior of $b_{x,I}$ in 1D and 2D is particular to the Gaussian and other functions that have simple derivatives. For example, this will not be the case for the 2D decaying exponential, f_2.

The dependence of $b_{x,I}$ on the other parameters, namely, the number of extrema, N_E, and ε_0 was also found to be similar in 2D to the behavior found in 1D. This is illustrated in Figs. 10 and 11.

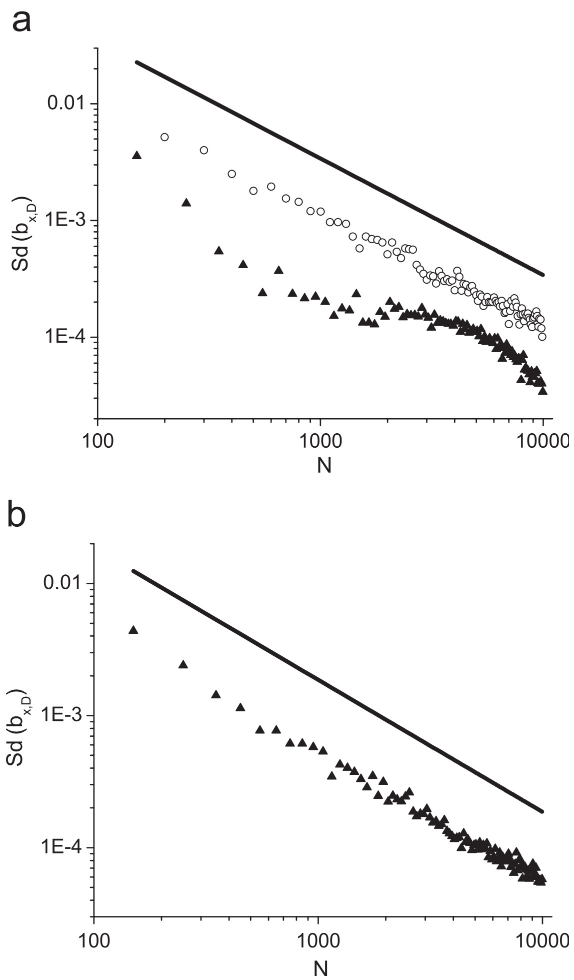

Fig. 8. The 2D analog of Fig. 4. The parameters are the same as in Fig. 4 and the functions are the 2D analogs with x replaced by r. Only the case of linear interpolation is shown. (a) $Sd(b_{x,D})$ for the 2D Gaussian (triangles) is compared with the 1D case (circles) and the theoretical estimate of Eq. (16) that here is the same for 1D and 2D (line). (b) $Sd(b_{x,D})$ for the 2D decaying exponential (triangles) compared to theory (line).

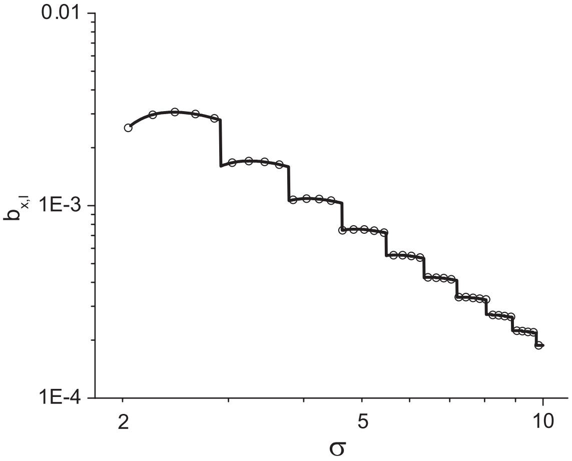

Fig. 9. The 2D analog of Fig. 5. The $b_{x,I}$ for the 2D Gaussian with the same parameters as in Fig. 5 (circles) is compared to the data of Fig. 5 (line).

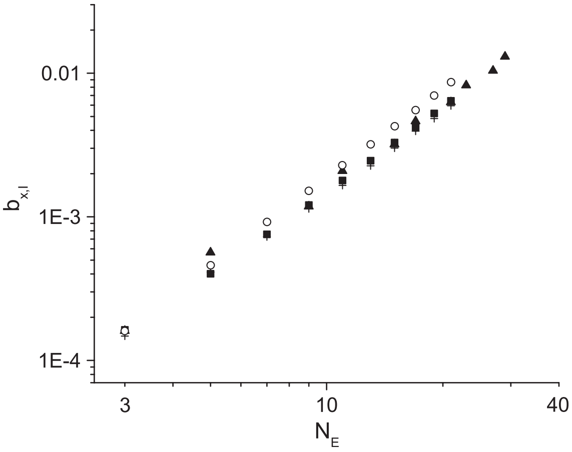

Fig. 10. The 2D analog of Fig. 6. Both the parameters and the symbols are the same as in Fig. 6 only that here the functions are the corresponding 2D analogs of the functions of Fig. 6. For comparison the behavior of the 1D cosine of Fig. 6, f_3, with $A = 40$ and $c = 1$ is also shown (circles).

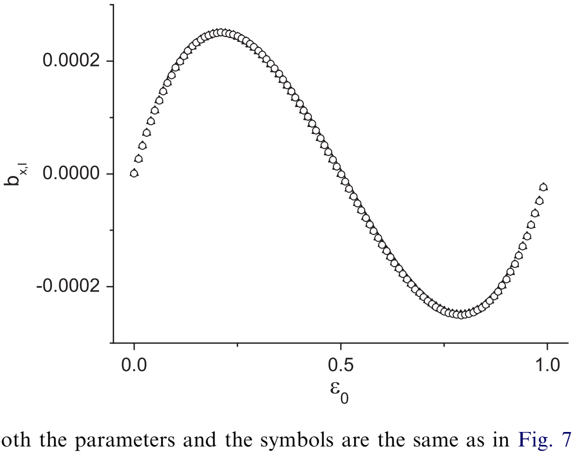

Fig. 11. The 2D analog of Fig. 7. Both the parameters and the symbols are the same as in Fig. 7 only that here the functions are the corresponding 2D analogs of the functions that were used in Fig. 7.

5. Tracking microbeads with the symmetry algorithm—experiment

We have compared the performance of the symmetry and cross-correlation algorithms for the case of latex beads (Polysciences) of 2.8 μm diameter. These were imaged on an inverted Olympus microscope (IX70) with a 100X objective (1.35 NA, oil immersion). Images were recorded using an analog CCD (Cohu 4912) and captured to computer with an 8-bit/pixel ($N = 256$) video card (Scion LG3-64 CCIR). The CCD has a 6.4×4.8 mm image area containing 752×582 pixels. The pixel size is 8.6×8.3 μm. The S/N ratio is advertised to be better than 56 dB at 0 gain and $\gamma = 1$. In our system, the pixel corresponds to an area of 82×78 nm of the sample. The beads were fixed to the glass bottom of the sample and we recorded 100 frames at 25 frames/s (see Fig. 12). We compute the distance between the centers of two beads, d_{12}, in each frame. In order to reduce the influence of the background, we limit the range of i and j in Eq. (1) such that $i^2 + j^2 \leqslant r^2$, where $r = 40$ pixels is about the size of the radius of the bead image. We find that the standard deviation of the distance, $Sd(d_{12})$, is similar for the two algorithms and that its values are within 20% of each other. $Sd(d_{12})$ varies in the range between 0.7 and 1.0 nm and is typically smaller when the cross-correlation algorithm is used. Using slightly defocused bead images for tracking (as in Fig. 12b) leads to larger differences between the performance of the two algorithms. In this case, the cross-correlation algorithm is up to 2.5 times more accurate than the symmetry algorithm. However, both algorithms give larger errors for such images.

For both focused and defocused bead images, the symmetry algorithm gives practically identical values for $Sd(d_{12})$ when either linear or cubic spline interpolation is used. This suggests that the discretization error, b_D, is dominant for such images, $b_D \gg b_I$.

In order to understand the experimental relevance of the discretization bias, we should compare its contribution to the tracking error to that from the other types of error. In particular, two types of noise need to be considered, namely, the dark noise, Δf_{dk}, and the photon counting noise, Δf_{ph}. While both the discretization error, Δf_D, and the dark noise in each pixel are independent of the intensity value, Δf_{ph}, is given by $\sqrt{N_{ph}}$, where N_{ph} is the number of photons that reaches the individual pixel in one frame.

To measure the number of photons we replaced the camera with a photodiode (UDT PIN-10D). Since the spectral responsivities of the camera and the photodiode are different, we performed the calibration in a 10 nm wide spectral window centered at $\lambda = 635$ nm using a band filter. In this spectral range, each intensity level in our CCD, ranging between 0 and 255, corresponds to about 355 photons. Varying the illumination intensity we find that $\langle f_i(N_{ph}) \rangle$ is linear with only 0.21% error in the slope. The average of f_i was taken over all the pixels in 20 consecutive frames. Next, we use the data to measure $Sd(f_i(N_{ph}))$. To eliminate the effects of background, we subtract consecutive frames and compute the standard deviation of the difference. The Sd of the difference is further divided by $\sqrt{2}$ and averaged over 19 frames. We find that $Sd(f_i(N_{ph})) = A_1\sqrt{N_{ph}} + B_1$ where $A_1 = 1.1 \pm 0.1$, $B_1 = 157 \pm 16$ and $N_{ph} \in (2500, 62\,500)$. The dark noise, corresponding to $N_{ph} = 0$, turns out to be larger than the value of B_1, namely, $Sd(f_i(N_{ph} = 0)) \simeq 208$. This could be due to the dependence of the read-out noise, Δf_{rd}, on N_{ph}. Since we cannot measure Δf_{rd} independently, in this discussion, we regard it as part of the dark noise, Δf_{dk}.

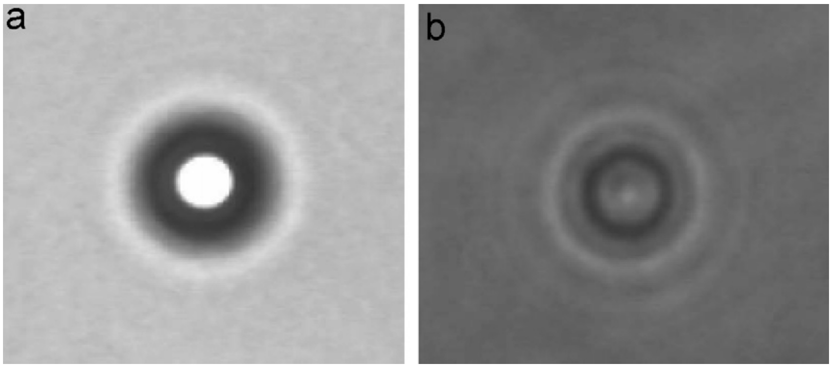

Fig. 12. Images of microbeads of 2.8 μm diameter that were used in the tracking experiment. (a) A focused bead image. Its center is saturated. This leads to the best tracking accuracy. (b) A defocused bead image.

To compare the three types of error it is convenient to express their magnitude in units of intensity on the 8-bit scale. For simplicity, we assume that the noise levels that we found do not depend on the wavelength. As discussed in Section 3.2, on this scale the discretization error is obtained as the standard deviation of a top-hat distribution over the $(-\frac{1}{2}, \frac{1}{2})$ interval, that is, $Sd(\Delta f_D) = \frac{1}{2\sqrt{3}} \simeq 0.29$. On the other hand, $Sd(\Delta f_{dk}) \simeq 0.59$ and the photon counting error will further increase that up to a maximum of $Sd(\Delta f_{ph,max}) + Sd(\Delta f_{dk}) \simeq 1.32$. Thus, for our CCD the three types of errors are of similar magnitude and none can be neglected. This prevents us from comparing the measured error with the one estimated by our analysis. Such comparison is further impeded by the fact that since Δf_{ph} depends on the intensity, Eq. (18) cannot be applied and needs to be generalized to include this situation. Such generalization will allow to minimize the tracking error given the characteristics of the CCD and the particle image.

6. Conclusions

We have presented a simple algorithm for symmetric particle tracking and have analyzed the behavior of the corresponding discretization error. The structure of the symmetry function was discussed and was used to deduce its behavior at the minimum. The discretization error was shown to be composed of a part that is due to the intensity discretization and another that is due to interpolation. Approximate formulas for these two components of the error were derived and compared to numerical results for representative functions. Finally, we applied the symmetry algorithm to bead images and compared the magnitude of the various sources of error.

The total error, $\mathbf{b}_T$, contains additional contributions from optical abberations and imperfections of the CCD, e.g. banding, interlace, clock jitter, etc. [26,27]. In a high quality, low noise, cooled CCD it should be possible to isolate the discretization error and compare its behavior to our estimates. In CCD's N is always a power of 2, $N = 2^q$, where typically $q = 8$, 12 or 16. Therefore, it might seem inappropriate to treat N as an arbitrary integer as was done in this study. However, often images cannot span the entire dynamic range of the camera and as a result a smaller, effective value of N, N_{eff}, needs to be considered.

The symmetry algorithm is conceptually not very different from the other tracking algorithms studied in Ref. [19]. It is therefore natural to expect that some of its features will also manifest in the other algorithms. In particular, the SAD algorithm [19,22] has a quite similar structure to that of the symmetry algorithm. While the former minimizes the absolute value of the difference between the images in consecutive frames, the latter does that for the two sides of the symmetry axis. In fact, the main difference between the SAD and symmetry algorithms is that the latter uses interpolation to obtain sub-pixel precision. Therefore, the behavior of b_D is expected to be similar in the two algorithms.

Acknowledgment

We thank I. Abdulhalim, J. Bechhoefer, I. Dinstein, L. Gheber and C. Kedem for useful discussions and suggestions. This research was supported in part by the Israel Academy of Science and Humanities (Grant no. 263/00).

References

[1] G. Gregoire, H. Chate, Y. Tu, Phys. Rev. E 64 (2001) 011902.

[2] T.G. Mason, D.A. Weitz, Phys. Rev. Lett. 74 (1995) 1250.

[3] A.R. Bausch, F. Ziemann, A.A. Boulbitch, K. Jacobson, E. Sackmann, Biophys. J. 75 (1998) 2038.

[4] M.A. Dichtl, E. Sackmann, Proc. Natl. Acad. Sci. USA 99 (2002) 6533.

[5] J. Gelles, B.J. Schnapp, M.P. Sheetz, Nature 331 (1988) 450.

[6] K. Svoboda, C.F. Schmidt, B.J. Schnapp, S.M. Block, Nature 365 (1993) 721.

[7] L.S. Barak, W.W. Webb, J. Cell Biol. 90 (1981) 595.

[8] L.S. Barak, W.W. Webb, J. Cell Biol. 95 (1982) 846.

[9] D. Gross, W.W. Webb, Biophys. J. 49 (1986) 901.

[10] R.N. Ghosh, W.W. Webb, Biophys. J. 66 (1994) 1301.

[11] T. Schmidt, G.J. Schutz, W. Baumgartner, H.J. Gruber, H. Schindler, Proc. Natl. Acad. Sci. USA 93 (1996) 2926.

[12] G.J. Schutz, H. Schindler, T. Schmidt, Biophys. J. 73 (1987) 1073.
[13] U. Kubitscheck, O. Kuckmann, T. Kues, R. Peters, Biophys. J. 78 (2000) 2170.
[14] M. Goulian, S.M. Simon, Biophys. J. 79 (2000) 2188.
[15] M.J. Saxton, K. Jacobson, Annu. Rev. Biophys. Biomol. Struct. 26 (1997) 373.
[16] R.J. Cherry, P.R. Smith, I.E. Morrison, N. Fernandez, FEBS Lett. 430 (1998) 88.
[17] A. Yildiz, J.N. Forkey, S.A. McKinney, T. Ha, Y.E. Goldman, P.R. Selvin, Science 300 (2003) 2061.
[18] J.C. Crocker, D.G. Grier, J. Colloid Interface Sci. 179 (1996) 298.
[19] M.K. Cheezum, W.F. Walker, W.H. Gullford, Biophys. J. 81 (2001) 2378.
[20] A. Kusumi, Y. Sako, M. Yamamoto, Biophys. J. 65 (1993) 2021.
[21] W.H. Guilford, R.W. Gore, Microvasc. Res. 50 (1995) 260.
[22] L.N. Bohs, B.H. Friemel, B.A. McDermott, G.E. Trahey, Ultrasound Med. Biol. 19 (1993) 751.
[23] G.M. Lee, A. Ishihara, K.A. Jacobson, Proc. Natl. Acad. Sci. USA 88 (1991) 6274.
[24] C.M. Anderson, G.N. Georgiou, I.E.G. Morrison, G.V.W. Stevenson, R.J. Cherry, J. Cell. Sci. 101 (1992) 415.
[25] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes, Cambridge University Press, New York, 1997.
[26] S. Inoue, K. Spring, Video Microscopy: The Fundamentals, Plenum Press, New York, 1997.
[27] T.C. Rigden, A.J. Soroka, L.A. Binns, Proc. SPIE Int. Soc. Opt. Eng. 5752 (2005) 1060.