Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Structural Biology 151 (2005) 182–195
Journal of Structural Biology
www.elsevier.com/locate/yjsbi

Feature point tracking and trajectory analysis for video imaging in cell biology

I.F. Sbalzarini *, P. Koumoutsakos

Institute of Computational Science, ETH Zürich, 8092 Zürich, Switzerland

Received 18 February 2005; received in revised form 1 June 2005
Available online 11 July 2005

Abstract

This paper presents a computationally efficient, two-dimensional, feature point tracking algorithm for the automated detection and quantitative analysis of particle trajectories as recorded by video imaging in cell biology. The tracking process requires no a priori mathematical modeling of the motion, it is self-initializing, it discriminates spurious detections, and it can handle temporary occlusion as well as particle appearance and disappearance from the image region. The efficiency of the algorithm is validated on synthetic video data where it is compared to existing methods and its accuracy and precision are assessed for a wide range of signal-to-noise ratios. The algorithm is well suited for video imaging in cell biology relying on low-intensity fluorescence microscopy. Its applicability is demonstrated in three case studies involving transport of low-density lipoproteins in endosomes, motion of fluorescently labeled Adenovirus-2 particles along microtubules, and tracking of quantum dots on the plasma membrane of live cells. The present automated tracking process enables the quantification of dispersive processes in cell biology using techniques such as moment scaling spectra.

Keywords: Single particle tracking; Feature point tracking; Tracker; SPT; Trajectory analysis; Moment scaling spectrum

1. Introduction

Techniques such as multi-color video microscopy and Single Particle Tracking (SPT) are becoming indispensable in cell biology. The quantitative analysis of the resulting trajectories provides important information about working mechanisms and structures in living cells. SPT (DeBrabander et al., 1985) has been used first for descriptive studies of plasma membrane protein and lipid diffusion (Anderson et al., 1992; Ghosh and Webb, 1994; Hicks and Angelides, 1995; Zhang et al., 1991). With the development and availability of new microscopy techniques such as confocal microscopy and Total Internal Reflection Microscopy (TIRFM) (Toomre and Manstein, 2001), it has become possible to classify modes of motion in live cells, determine diffusion coefficients of single molecules (Goulian and Simon, 2000), or measure the step displacements of molecular motors such as kinesin (Gelles et al., 1998). Descriptions and overviews of the employed analysis methods are available in the review by Saxton and Jacobson (1997).

Video microscopy of fluorescently labeled virus particles transported on cell surfaces and into internal organelles led to the pioneering study of Pelkmans et al. (2001, 2002). Using frames from videos, they visualized and analyzed many of the key steps in the early pathway of the caveolar entry of SV40 into live cells (Pelkmans et al., 2001). This analysis was performed tracking by hand the individual particles, a procedure that becomes impossible when one needs to analyze the multitude of trajectories available by today's video techniques.

* Corresponding author.
E-mail addresses: sbalzarini@inf.ethz.ch (I.F. Sbalzarini), petros@inf.ethz.ch (P. Koumoutsakos).

1047-8477/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jsb.2005.06.002

The feature point tracking problem consists of detecting images of particles in a digital video sequence and linking these detections over time to follow the traces of individual particles. Applications of feature point tracking are numerous in several fields of science and technology such as fluid mechanics (e.g., particle imaging velocimetry, particle tracking velocimetry Wereley et al., 2002), computer vision (e.g., road following Morgan et al., 1990, human limb tracking Lerasle et al., 1999), navigation (e.g., vehicle navigation Sanchiz and Pla, 1999), material science (e.g., colloids Crocker and Grier, 1996), and biology (e.g., membrane protein and lipid diffusion Anderson et al., 1992; Fujiwara et al., 2002; Ghosh and Webb, 1994; Hicks and Angelides, 1995; Zhang et al., 1991). A number of specialized, often application-specific, algorithms, and computer programs is available (Chetverikov and Verestóy, 1999; Cheezum et al., 2001; Vallotton et al., 2003). Most of them make use of a priori knowledge about the physics of the problem to construct effective and robust feature point tracking procedures. Some of them are very accurate, but computationally intense, which prohibits tracking of long video sequences.

Biological applications, as those mentioned above, often involve the tracking of objects whose type of motion may not be known explicitly in advance. In these cases, the tracking task is hindered by the absence of a suitable mathematical model, by the possible stochastic character of the motion, or by trajectories entailing several modes of motion (e.g., smooth and non-smooth parts). Most biological applications only require the two-dimensional case since the motion either is two-dimensional (e.g., on the plasma membrane) or is observed using "two-dimensional" microscopy techniques such as TIRFM (Toomre and Manstein, 2001) or confocal microscopy.

In this article, the automated computation of trajectories is developed under the basic assumptions of small feature points (compared to the length scale of background variations), limited speed, and short occlusions. The present algorithm is self-initializing and capable of handling occlusion, exit, and entry. It is in the same functional class as the IPAN tracker introduced by Chetverikov and Verestóy (1999), except that the present work makes no assumptions about the smoothness of the trajectories. The present algorithm is fast and efficient, while at the same time having accuracy and precision that are comparable to far more computationally intensive algorithms. The algorithm relies on a minimum set of assumptions, reduced prior knowledge of the physical process, and a small set of user-defined parameters. It is suitable for tracking of long videos of mobile objects such as viruses on the plasma membrane, fast-directed motion such as trafficking along microtubules, and particles with strong intensity fluctuations such as quantum dots (Qdots).

The automated detection of trajectories from video sequences provides us with a wealth of information that can be exploited to quantify further the particle motions. The analysis of two-dimensional trajectories is a well established technique to determine diffusion coefficients or transport velocities (Qian et al., 1991). In the present work, we employ Moment Scaling Spectra (MSS) (Ferrari et al., 2001), first introduced in fluid mechanics to quantify dispersion processes. This technique is shown to provide a systematic characterization of particle trajectories, enabling a rigorous quantification of biological dispersion processes.

2. Feature point tracking algorithm

2.1. Definitions

The following problem definition and terminology will be used throughout this article: we consider physical *particles* that are mobile in a two-dimensional plane. Their motion is *observed* using imaging equipment and a digital (CCD) camera which generates a sequence of digital images at discrete time points. We call this sequence a *movie* and an individual image from it a *frame*. In each frame, the images of the particles are visible as *feature points* (or *points*). The goal is to approximately reconstruct the motion of the observed particles. Such a reconstruction consists of an ordered series of point locations over the recording time points of the individual frames and is called a (discrete) *trajectory*. To generate the trajectories, the *feature point tracking* algorithm has to perform two distinct steps: first it has to *detect* the feature points in every frame and then it has to *link* these point detections into trajectories. If a point is detected where there is none, we call it a *false detection*. The term *"spurious detection"* on the other hand refers to a correctly detected point where there was no particle of the desired kind in the real scene. Finally, linking two points that are not images of the same physical particle is called a *false link*. If a trajectory does not extend through the whole image sequence it is called *incomplete*. This manuscript describes and tests an algorithm for *feature point tracking*, which is a sub-problem of *particle tracking* and does not include treatment and analysis of the physical system under observation and the used imaging equipment.

2.2. Feature point detection

The algorithm is initialized by determining the global[1] minimum $I_{\min}$ and maximum $I_{\max}$ of all intensity

[1] That is, across all the frames of the movie rather than within each frame individually.

values occurring in the movie. All pixel intensity values I are then normalized as $(I-I_{\min})/(I_{\max}-I_{\min})$. The use of global extrema preserves intensity variations across frames, serving as an important source of information in the linking step. The feature point detection consists of four steps:

(1) image restoration;
(2) estimation of the point locations;
(3) refinement of the point locations;
(4) non-particle discrimination.

The implemented algorithm has as a starting point the work by Crocker and Grier (1996) for the detection of gold colloids in micrographs. In the following, the normalized frame image at observation time t is represented as a matrix $A^t(x,y)$ of floating point intensity values between 0 and 1. The integer coordinate $x = 1, \ldots, N_x$ is the pixel row index and $y = 1,\ldots, N_y$ is the pixel column index.

The *image restoration* corrects for imperfections in the frame images. There are two different effects accounted for: (1) long-wavelength modulations of the background intensity due to non-uniform sensitivity among the camera pixels or uneven illumination, and (2) discretization noise from the digital camera. The former is straightforward to correct for since we assume the feature points to be small compared to background variations and thus well separated in spatial frequency. The background is removed by a boxcar average over a square region with extent of $2w+1$ pixel:

$$A^t_w(x,y) = \frac{1}{(2w+1)^2}\sum_{i=-w}^{w}\sum_{j=-w}^{w} A^t(x+i,y+j), \tag{1}$$

where the user-defined parameter w is an integer larger than a single point's apparent radius but smaller than the smallest inter-point separation. The camera discretization noise is modeled as uniformly Gaussian with a correlation length of $\lambda_n = 1$ pixel. The de-noising filter thus consists of a convolution of the image A^t with a Gaussian surface of revolution of half width λ_n (Crocker and Grier, 1996):

$$A^t_{\lambda_n}(x,y) = \frac{1}{B}\sum_{i=-w}^{w}\sum_{j=-w}^{w} A^t(x+i,y+j)\exp\left(-\frac{i^2+j^2}{4\lambda_n^2}\right) \tag{2}$$

with normalization

$$B = \left[\sum_{i=-w}^{w}\exp(-(\mathrm{i}^2/(4\lambda_n^2)))\right]^2. \tag{3}$$

Both Eqs. (1) and (2) amount to convolving the image with kernels of support $2w+1$. The steps are thus combined and the final image restoration consists of a convolution of the original frame image with the kernel

$$K^w(i,j) = \frac{1}{K_0^w}\left[\frac{1}{B}\exp\left(-\frac{i^2+j^2}{4\lambda_n^2}\right) - \frac{1}{(2w+1)^2}\right]. \tag{4}$$

The normalization constant

$$K_0^w = \frac{1}{B}\left[\sum_{i=-w}^{w}\exp\left(-\frac{\mathrm{i}^2}{2\lambda_n^2}\right)\right]^2 - \frac{B}{(2w+1)^2} \tag{5}$$

allows comparison among images filtered with different values of w. The filtered image after restoration is given by:

$$A^t_f(x,y) = \sum_{i=-w}^{w}\sum_{j=-w}^{w} A^t(x-i,y-j)K^w(i,j). \tag{6}$$

To perform the convolution, the image is temporarily padded to size $(N_x+2w)\times(N_y+2w)$ by repeating the first and last row and column each w-fold outwards. Negative pixel values generated by the convolution are reset to 0. They are an artifact of the approximation of the camera noise by a Gaussian distribution, which breaks down at small intensity levels.

Estimating the feature point locations is done by finding local intensity maxima in the filtered image A^t_f. A pixel is taken as the approximate location of a point if no other pixel within a distance of w is brighter and if its intensity is in the upper rth percentile of intensity values of the current frame image. The intensity percentiles are determined on a per frame basis to be robust against possible global drift in image intensity over time, e.g., due to unspecific bleaching of the observed particles. The local maximum selection is implemented as a grayscale dilation (Jain, 1986) followed by the selection of all pixels that have the same value before and after the dilation. If such a pixel is in the upper rth percentile of intensity values, it is taken as the candidate location of a point. The local maximum selection of point centers suffers from two deficiencies: (1) it is unable to reject noise, which leads to errors in the location estimate, and (2) it will include spurious detections such as random bright points in the background of the image or images of particle aggregates. This makes both a refinement of the detected locations and a subsequent non-particle discrimination necessary.

Refining the point locations will reduce the standard deviation of the position measurement. Other information gathered in the process can furthermore be reused later to reject spurious detections. The assumption is that the found local maximum of a point p at $(\hat{x}_p,\hat{y}_p)$ is near the true geometric center (x_p,y_p) of the particle. An approximation of the offset is given by the distance to the brightness-weighted centroid in the filtered (to

reduce noise-induced positioning errors) image A^t_f (Crocker and Grier, 1996):

$$\begin{bmatrix}\varepsilon_x(p)\\ \varepsilon_y(p)\end{bmatrix} = \frac{1}{m_0(p)} \sum_{i^2+j^2\leqslant w^2} \begin{bmatrix} i \\ j \end{bmatrix} A^t_f(\hat{x}_p + i, \hat{y}_p + j). \tag{7}$$

The normalization factor $m_0(p)$ is the sum of all pixel values over a feature point p, i.e., its intensity moment of order 0:

$$m_0(p) = \sum_{i^2+j^2\leqslant w^2} A^t_f(\hat{x}_p + i, \hat{y}_p + j). \tag{8}$$

The location estimate is refined as: $(\tilde{x}_p, \tilde{y}_p) = (\hat{x}_p + \varepsilon_x(p), \hat{y}_p + \varepsilon_y(p))$. If either $|\varepsilon_x(p)|$ or $|\varepsilon_y(p)|$ is larger than 0.5 pixel, the candidate location $(\hat{x}_p, \hat{y}_p)$ is accordingly moved by one pixel and the refinement recalculated.

The *non-particle discrimination* should reject spurious detections such as non-specific signals, dust, or particle aggregates. The implemented classification algorithm after Crocker and Grier (1996) is based on the intensity moments of orders 0 and 2. The 0*th* order moment of each point p has already been calculated in the previous step. The second-order intensity moment is computed as:

$$m_2(p) = \frac{1}{m_0(p)} \sum_{i^2+j^2\leqslant w^2} (i^2 + j^2) A^t_f(\hat{x}_p + i, \hat{y}_p + j). \tag{9}$$

The underlying assumption is that the majority of the detected observations correspond to correct particles such that they form a dense cluster in the (m_0,m_2)-plane. Larger and dimmer or brighter structures such as aggregates or accumulations will have different intensity momenta and fall outside of the main cluster. They are identified by having each point p "carry" a 2D Gaussian

$$P_p(m_0, m_2) = \frac{1}{2\pi\sigma_0\sigma_2 N_t} \times \exp\left(-\frac{(m_0 - m_0(p))^2}{2\sigma_0} - \frac{(m_2 - m_2(p))^2}{2\sigma_2}\right) \tag{10}$$

with standard deviations σ_0 and σ_2, and N_t the total number of detected points in the current frame. The contributions of all other points $q \neq p$ are summed for each point p at its location, giving a score

$$S_p = \sum_{q\neq p} P_q(m_0(p), m_2(p)). \tag{11}$$

Every point detection having a score S_p above a certain user-provided threshold T_s is considered as an observation of a "true" particle, all others are discarded.

Notice that the standard deviations σ_0 and σ_2 define the length scale of the clustering and can be chosen such as to normalize the cluster widths. Let $I_{\max}$ be the maximum intensity in the movie; $I_{\max} = 1$ if the images are normalized as described earlier. We then have the bounds $0 \leqslant m_0 < I_{\max}\pi w^2$ and $0 \leqslant m_2 < I_{\max}\pi w^4/2$, which can be used to estimate values for σ_0 and σ_2. In our experience, a value of about $0.1 I_{\max}\pi w^2$ seems to be a good choice. Fig. 1 illustrates the non-particle discrimination clustering applied to a confocal image of fluorescently labeled Polyoma virus particles on a PTK2 cell. The image shows a confocal slice through the cell and thus contains observations of virus particles

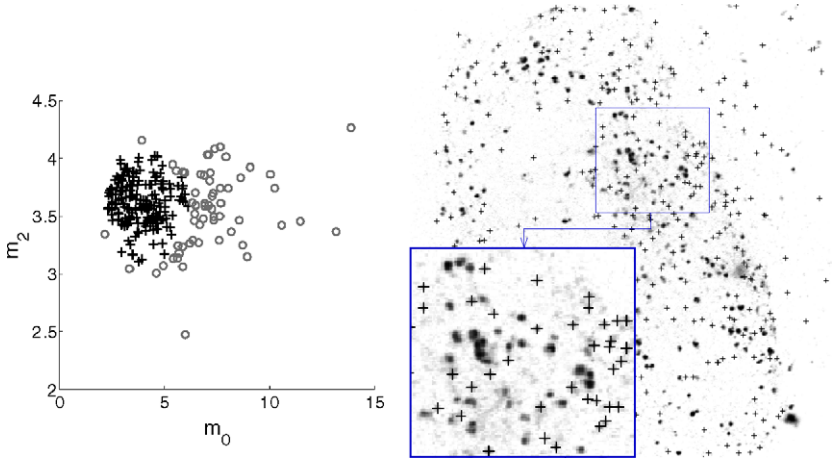

Fig. 1. Left panel: example of the non-particle discrimination clustering in the (m_0,m_2)-plane. Each symbol represents one detected feature point. The clustering with $\sigma_0 = \sigma_2 = 0.1$, and $T_s = 2.0$ (images normalized to $I_{\max} = 1$) classifies the points marked by a plus symbol as "true" particles. Points outside the cluster are marked by circles and are classified as spurious detections. Right panel: confocal image of fluorescently labeled Polyoma virus particles on a PTK2 cell (image intensities are inverted for printing purposes). The confocal slice contains both extracellular and intracellular regions. Already internalized virus particles are packed together in endocytic organelles that appear as larger fluorescent structures and are to be excluded from the trajectory linking. The result of the clustering shown in the left panel is illustrated with plus symbols marking "true" particles. Inset shows enlargement as indicated.

both on the plasma membrane and in the interior of the cell. The clustering is used to discard virus particles packed together in endocytic organelles, allowing analysis of individual free particles.

2.3. Trajectory linking

The feature point detection algorithm is applied to each frame image A^t and yields a set of T (total number of frames in the movie) matrices $C^t \in \mathbb{R}^{N_t \times 2}$ with rows $[\tilde{x}_p, \tilde{y}_p]_{p=1}^{N_t}$, where N_t is the total number of points detected in frame t.

The linking algorithm identifies points corresponding to the same physical particle in subsequent frames and links the positions $\{C^t\}_{t=1}^{T}$ into trajectories. This involves finding a set of associations between the point location matrices $\{C^t\}_{t=1}^{T}$ such that a cost functional is minimized. The present implementation is based on a particle matching algorithm (Dalziel, 1992, 1993a,b) using a graph theory technique (Hichcock, 1941) to determine optimal associations between two sets. This algorithm is extended so that each linking step may consider several frames to account for particle occlusion.

Let $\mathscr{P}$ represent the set of points p_i, where $i = 1,\ldots, N_t$, in frame t and R a user-defined integer parameter specifying how many future frames are to be considered. For all sets $\mathscr{Q}_r$, $r = 1,\ldots, R$, of points q_j, where $j = 1,\ldots, N_{t+r}$, in frame $t + r$ an association matrix G_r^t is defined:

$$G_r^t(i,j) = g_{ij} = \begin{cases} 1 & \text{if } p_i \text{ in frame } t \text{ and } q_j \text{ in frame } t+r \text{ are produced by the same physical particle,} \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We assume that there is always exactly one physical particle producing a single point detection. Note that this is a limiting assumption since particles could in principle coalesce or come so close that they are indistinguishable by the used imaging equipment, giving raise to one single point observation.

To allow the number of points to vary between frames, i.e., $N_t \neq N_{t+r}$, every association matrix is augmented with both a row g_{0j} and a column g_{i0} for *dummy particles* at times t and $t + r$, respectively. Linking a point to the dummy means that the corresponding particle disappeared from the observed part of the scene between frames t and $t + r$, and linking the dummy to a point means that the corresponding particle newly appeared. This leads to the following topology constraint on the matrices G_r^t:

Every row $i > 0$ of G_r^t and every column $j > 0$ of G_r^t must contain exactly one entry of value 1, all others are 0. Row 0 and column 0 are allowed to contain more than one entry of value 1.

To find an optimal set of links g_{ij}, we need to define the cost functional to be minimized. To be able to use the efficient solution algorithm based on the transportation problem (Dalziel, 1993b; Hichcock, 1941), this functional needs to be linear in the association variables g_{ij} and may thus be written as the linear combination

$$\Phi = \sum_{i=0}^{N_t} \sum_{j=0}^{N_{t+r}} \phi_{ij} g_{ij}, \tag{13}$$

where ϕ_{ij} represents the cost of associating point p_i in frame t with point q_j in frame $t + r$. The definition of ϕ typically includes the point positions, characteristics, or (if available) temporal and spatial knowledge about the physics of the process. For the above functional to be linear, ϕ itself must not depend on the association variables g_{ij}. In our case, we use the quadratic distance between p_i, $i > 0$, and q_j, $j > 0$, as well as the quadratic differences in the intensity moments of order 0 and 2, thus:

$$\phi_{ij} = (\tilde{x}_{p_i} - \tilde{x}_{q_j})^2 + (\tilde{y}_{p_i} - \tilde{y}_{q_j})^2 + (m_0(p_i) - m_0(q_j))^2 + (m_2(p_i) - m_2(q_j))^2 \tag{14}$$

for $i,j > 0$. The cost of linking a point to one of the dummy particles $i = 0$ or $j = 0$ is set equal to: $\phi_{0j} = (rL)^2$, $j > 0$, and $\phi_{i0} = (rL)^2$, $i > 0$. This effectively places a limit to the allowed cost for point associations since no association of cost larger than $(rL)^2$ will occur between regular points because the dummy association would be more favorable. The parameter L is specified by the user and represents the maximum distance a point is allowed to travel between two subsequent frames when its intensity moments remain constant. To speed up the linking process, all costs $\{\phi_{ij}:\phi_{ij} > (rL)^2\}$ are set to ∞ and the corresponding g_{ij} will never be considered in the following.

2.3.1. Initialization

The linking process starts by creating an arbitrary set of associations g_{ij} that satisfies the topology constraint. Any valid association matrix G_r^t is acceptable since the linear nature of Φ ensures that the minimum of the objective function is unique (Dalziel, 1993b). Learned choice of the initial associations can significantly reduce the number of iterations needed in the subsequent optimization process. The initial set of links is thus determined as follows: For each pair of frames (t,r), $r = 1,\ldots, R$, the association matrix G_r^t is initialized by assigning each point in frame t its nearest neighbor, using ϕ as the distance measure, in frame $t + r$ that is not already assigned to some other point. This means that for every given $i = I$, $j = J$ is chosen such that ϕ_{IJ} is the minimum of all ϕ_{Ij} for which no other g_{iJ} is already set to 1. This g_{IJ} is then set to 1. If no such minimum is found, the point is linked to the dummy, i.e., g_{I0}

is set to 1. After having done this for all the points p_i, every J for which no g_{iJ} is set is determined and the corresponding g_{0J} is set to 1. This initialization generates a matrix G_r^t that fulfills the topology constraint. For low point densities, this initial solution is already very close to optimal since only few conflicts occur (i.e., the association that would have had the lowest cost was already blocked by another one). To cope with regions of high point density, the association matrix is iteratively optimized.

2.3.2. Optimization

For each iteration, we scan through all g_{ij}, including the dummy particles, that are equal to 0 *and* have finite associated cost ϕ_{ij}. For these we determine the *reduced cost* of introducing that association into the matrix. The reduced cost is calculated from the elementary costs ϕ for $i,j > 0$ by considering a zero association g_{IJ}, $I,J > 0$. Let $g_{IL} = 1$ and $g_{KJ} = 1$, since every row and column must contain a 1 according to the topology constraint. Now if g_{IJ} was to be set to 1, then g_{IL} and g_{KJ} must turn 0, otherwise points p_I and q_J would be in two places at once. Further, as point detections $i = K$ and $j = L$ must be related to some physical particle, it is necessary to set $g_{KL} = 1$. The *reduced cost* of setting g_{IJ}, $I,J > 0$, to 1 thus is:

$$z_{IJ} = \phi_{IJ} - \phi_{IL} - \phi_{KJ} + \phi_{KL}, \qquad I,J > 0. \tag{15}$$

If the reduced cost z_{IJ} is negative, introducing the association g_{IJ} into the solution is favorable, i.e., will decrease the cost functional Φ. In the case of a newly appearing particle, i.e., the association under consideration is at g_{0J} for some $J > 0$, only the 1 in the same column at g_{KJ}, $K > 0$, is turned into a 0 and the dummy entry g_{K0} is set to 1. The reduced cost for an appearing particle thus is:

$$z_{0J} = \phi_{0J} - \phi_{KJ} + \phi_{K0}, \qquad J,K > 0, L = 0. \tag{16}$$

For a disappearing particle we similarly have:

$$z_{I0} = \phi_{I0} - \phi_{IL} + \phi_{0L}, \qquad I,L > 0, K = 0, \tag{17}$$

setting g_{I0}, $I > 0$, to 1, turning g_{IL}, $L > 0$, from 1 to 0, and setting the dummy g_{0L} to 1 as well. The special case $I = J = 0$ is set to $z_{00} = 0$. After calculating the reduced costs $\forall\{(i,j)\colon g_{ij} = 0 \wedge \phi_{ij} < \infty\}$, the g_{IJ} which corresponds to the most negative reduced cost $z_{IJ} = \min_{i,j} z_{ij}$ is set to 1, the corresponding g_{IL} (if $I \neq 0$) and g_{KJ} (if $J \neq 0$) to 0 and g_{KL} to 1. All the reduced costs are then re-calculated and the iteration is repeated until $z_{ij} \geqslant 0\ \forall(i,j)$, which means that the optimal set of associations, with cutoff L, between frames t and $t + r$ has been found.

After doing so for all $r = 1, \ldots, R$ and a fixed specific t, all points in C^t that have been linked to the dummy particle in C^{t+1} are closely analyzed to re-connect broken trajectories caused, e.g., by particle occlusion, a sensitive non-particle discrimination, or a particle being close to the intensity percentile threshold. For each such point p_i in frame t, all association matrices G_r^t, $r = 2, \ldots, R$ are scanned for valid associations to non-dummy points. If there are such associations, the one that has the smallest reduced cost is accepted and the corresponding point detections are linked.

Repeating the whole procedure for every frame t leads to an optimal (in the sense of the chosen cost functional Φ) linking of the detected point locations into trajectories over time. The computational cost of this linking algorithm formally scales as $\mathcal{O}(R(N^2 - N))$ and the algorithm needs $\mathcal{O}(RN^2)$ memory. Associations between well separated particles are however initially marked by an infinite cost and are never considered during optimization. This greatly improves the computational efficiency. The number of possible associations with finite cost values ϕ_{ij} is greater than or equal to $\max(N_t, N_{t+r})$, but much lower than $(N_t+1)(N_{t+r}+1)$, depending on the actual distribution of particles. In practice, the computational time for this algorithm increases only slightly more rapidly than $\mathcal{O}(RN)$ and R is usually small (1, 2, or 3). In fact, the trajectory linking takes less time than the feature point detection in most practical applications. A typical optimization of the association matrix G_r^t needs in the order of 10 iterations until the optimum set of links is found.

2.4. Computer implementation

The algorithm is implemented in ANSI C using a client-server model. The communication between the server and the clients is controlled by a simple packet-based protocol, which is directly built upon TCP/IP. This makes it possible to access the server from any remote computer that provides access to the network.

The *server* application essentially consists of two parts: communication and point tracking. The point tracking part provides an Application Programming Interface (API) that is used by the communication part. This API implements the algorithm described so far plus a set of functions for setting the tracking parameters, submitting a list of images, and retrieving the results. The communication part connects the tracker API with multiple, potentially concurrent clients. This is realized by multi-threading under the Windows operating system and multi-processing under Linux.

The *client* application provides an implementation of the communication protocol and a user interface. It sets the user-defined parameters of the tracking server, reads and submits an image sequence either as a series of TIFF images or directly from an MPEG-1 movie file, and receives the resulting trajectories from the server. Both a text-mode and a graphical client are available. The former is provided for efficient batch operation, e.g., use from scripts or other programs, the latter provides ease of use and assisted parameter choices, and

directly allows to inspect the final result of a tracking job and analyze the trajectories with regard to their motion properties, diffusion constants (Qian et al., 1991), or MSS (Ferrari et al., 2001). The graphical client can directly export processed data and diagram plots. Both clients have been tested on Mac OS X, Windows, and Linux operating systems.

3. Benchmarks

The quality of the feature point detection is evaluated using synthetic frame sequences of moving point blobs. This method of evaluation is preferred over the common experimental practice of tracking a stationary/fixed particle and use the variance of the detected point positions as a measure of tracking quality. The true accuracy of the algorithm is given by its bias (Cheezum et al., 2001), which cannot be estimated unless the precise and correct relative position of the particle with respect to the elements of the imaging system is known. The only way to achieve such conditions is by using numerical simulations.

A good tracking algorithm has to meet two independent measures of quality: it should minimize determinate errors resulting from inaccuracies inherent to the algorithm and it should minimize indeterminate errors from measurement fluctuations and imaging noise. While determinate errors will systematically bias the position detections toward incorrect values, indeterminate errors fluctuate randomly. Following the terminology of Cheezum et al. (2001), we will refer to the measure of determinate errors as *accuracy* and to the one of indeterminate errors as *precision*.

Both accuracy and precision are estimated for a moving point source at different signal-to-noise ratios (SNR) and pixel displacements per frame (Δx). Synthetic frames are created using particles moving on a horizontal straight line at constant speed Δx pixel/frame, cf. Fig. 3. Observation is simulated by centering a 2D Gaussian blob

$$I(x,y) = I_0 \cdot \exp\left(-\frac{(x-x_p)^2 + (y-y_p)^2}{4\sigma^2}\right) \tag{18}$$

of standard deviation $\sigma = 1$ pixel (Thompson et al., 2002) at the current particle location (x_p,y_p) and sampling its value at the center of all pixels, $x = \pm 1/2,\ \pm 3/2,\ \pm 5/2,\ldots$, $y = \pm 1/2,\ \pm 3/2,\ \pm 5/2,\ldots$. Gaussian blobs are used as an approximation to (1) the sinusoidal intensity distribution of radially emitting spherical beads and (2) the square Bessel point spread function of a sub-resolution particle imaged using a microscope. To model different SNR, a background (black) level of $b = 10$ is added to all pixels and the peak intensity v of the blobs is varied by setting $I_0 = v - b$ before adding the blobs to the images. For the noise model, we assume that the images are acquired using a digital CCD camera. Such cameras produce Poisson-distributed pixel noise due to the discrete photoelectron counting (Ryan et al., 1990). Pixel noise is thus simulated by replacing the intensity value I of each pixel with a random number from a Poisson distribution of expectation value $\lambda = I$. Fig. 2 illustrates the effect of such noise on a simulated point blob. All random numbers are generated independently for every trial and frame. The resulting frame images (cf. Fig. 3, left panel) are stored as unscaled 16-bit TIFF files.

The SNR is calculated as the difference in expected intensity levels between the particle points v and the

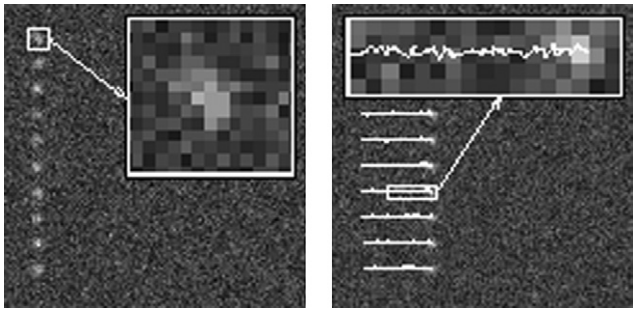

Fig. 3. Example benchmark tracks. Each test case consists of an image sequence of 100 frames of 10 moving simulated points, yielding 1000 independent displacement measurements â with known exact values $a = \Delta x$. First (left panel) and last (right panel) frame of an example with $\Delta x = 0.27$ pixel, peak level $v = 23.9$, and background level $b = 10$ (SNR = 2.846) are shown with lines depicting trajectories as reconstructed by the presented tracking algorithm. All 10 trajectories are of full length 100. Insets show enlargements as indicated.

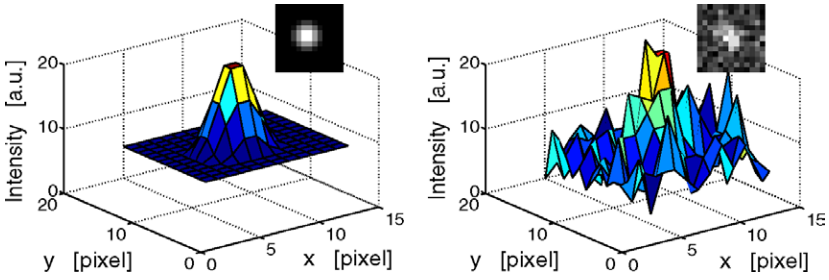

Fig. 2. Example of a simulated particle observation before (left panel) and after (right panel) addition of Poisson noise. Inset images show the particle images whose pixel intensity distributions are depicted in the surface plots below. The example shown uses a peak level of $v = 23.9$ and a background level of $b = 10$, thus having an SNR of 2.846.

background b, divided by the noise level σ_n on the particles. For the employed Poisson noise this is $\sigma_n = \sqrt{v}$ and thus:

$$\mathrm{SNR} = \frac{v - b}{\sqrt{v}}. \tag{19}$$

Note that this is the most conservative definition of SNR possible. Using the noise level of the background would lead to much larger values. These would be inappropriate (Cheezum et al., 2001) since the stronger noise on the bright blobs is the one that actually influences the feature point detection and causes its inaccuracy. The peak pixel levels used in the present benchmark cases are given in Table 1 along with the corresponding resulting SNR values according to Eq. (19).

Accuracy and precision of the algorithm are quantified for different SNR and Δx using, respectively, the track *bias*

$$\mathrm{bias} = \langle \hat{a} - a \rangle, \tag{20}$$

and its *standard deviation*

$$\sigma = \langle (\hat{a} - \langle \hat{a} \rangle)^2 \rangle^{1/2}. \tag{21}$$

Table 1
Peak pixel levels v and resulting SNR used for the cases in Fig. 4. The background level is fixed at $b = 10$

Peak level v	SNR
15	1.291059
18.58	1.990510
23.9	2.846111
28.73	3.494379
38.1	4.556798
60.8	6.516668
97	8.832892
154.7	11.632132
246.6	15.067460
393.3	19.326731
627.1	24.642859
1000	31.306549

Here, $\langle \cdot \rangle$ denotes the ensemble average over independent trials, $\hat{a}$ the reconstructed particle displacements from the tracking algorithm, and a the actual exact displacements. The tracking algorithm runs on a 3.06 GHz Intel Pentium 4 server, taking less than 1s to track 1000 independent displacements.

Fig. 3 shows both the first and the last frame at $\mathrm{SNR} = 2.85$. The trajectories as reconstructed by the tracking algorithm are shown as solid lines in the right panel. The bold circles in Fig. 4 show the results for accuracy and precision versus SNR for a fixed displacement of $\Delta x = 0.27$ pixel. Fig. 5 shows bias and standard deviation versus the magnitude of the actual particle displacement per frame between 0 and 1 pixel in steps of 1/11 pixel for a fixed SNR of 31.3.

The critical SNR for reaching an accuracy better than 0.1 pixel is around 4.2 for the present algorithm, indicating good capability to handle noisy images. The precision σ is better than 1 pixel for all SNR larger than 1.3, cf. Table 2. For SNR better than 7.5, both σ and bias are below 0.1 pixel.

The presented algorithm shows about the same accuracy as the more complex and computationally intense Gaussian fit and cross-correlation methods, while having better precision. The smooth and monotonic decay of both bias and standard deviation with increasing SNR are favorable properties of the present method and the bias is virtually constant (and low) for all step displacements $\Delta x > 0.1$ pixel. The fact that the present algorithm avoids fitting a specific point spread function shape to the blobs in the frame images not only results in faster execution speed, but also renders it more general with respect to size and shape of the tracked objects.

In a second test, the simulated points are moving along straight lines of random angular orientation, thus exhibiting truly two-dimensional motion. Trajectories can intersect and points can exit the image, in which case they reappear on the opposite side (periodic boundary

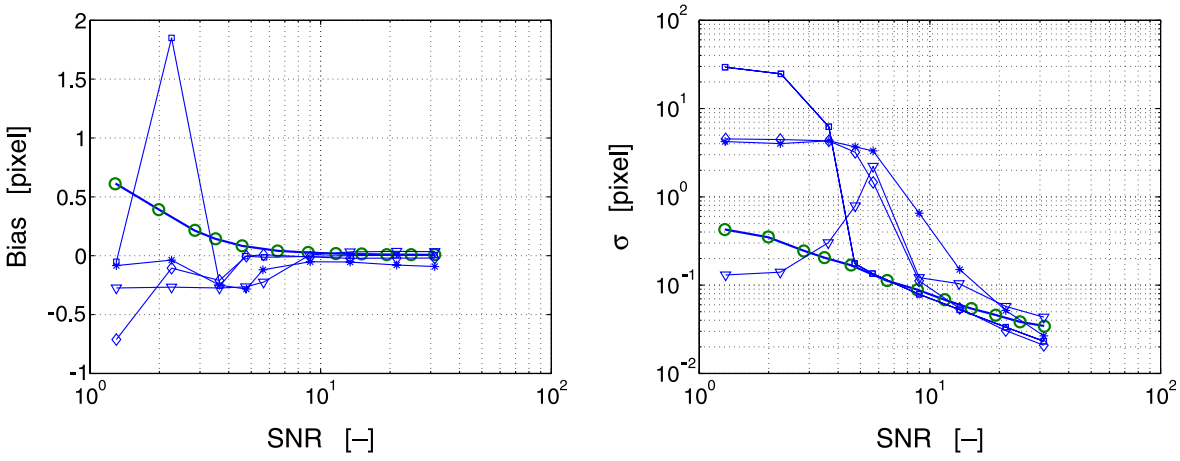

Fig. 4. Left panel: bias versus SNR for a point blob moving at 0.27 pixel/frame. Right panel: standard deviation versus SNR for the same cases. Each point is averaged from 1000 independent measurements. The present algorithm (bold circles) is compared to the data of Cheezum et al. (2001): Gaussian fit (squares), centroid (triangles), sum of absolute differences (stars), cross-correlation (diamonds).

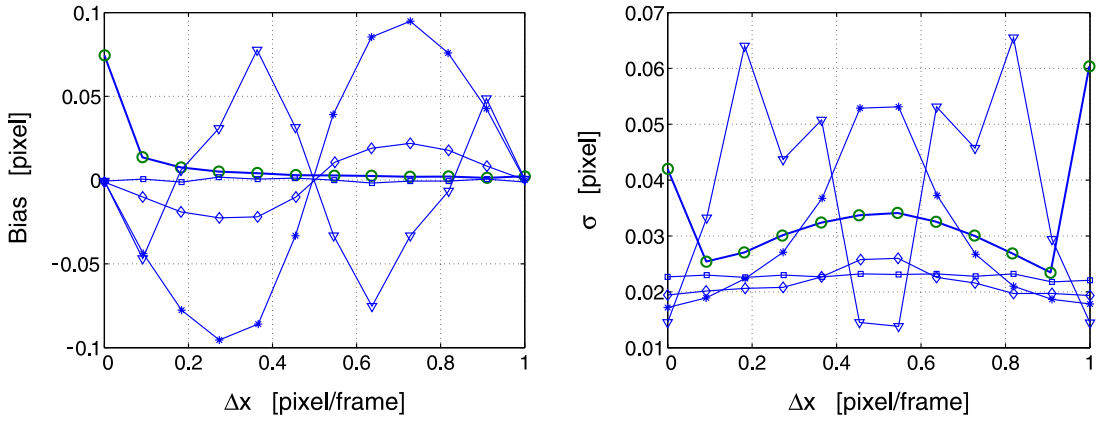

Fig. 5. Left panel: bias versus actual distance moved per frame for a point blob at SNR=31.3. Right panel: standard deviation versus actual distance moved per frame for the same cases. Each point is averaged from 1000 independent measurements. The present algorithm (bold circles) is compared to the data of Cheezum et al. (2001): Gaussian fit (squares), centroid (triangles), sum of absolute differences (stars), and cross-correlation (diamonds).

Table 2
Estimated SNR at which the bias drops below 0.1 pixel and σ below 1 pixel. Comparison of the present algorithm with the ones tested by Cheezum et al. (2001)

Algorithm	$SNR_{0.1bias}$	$SNR_{1.0\sigma}$
Present work	4.2	<1.3
Gaussian fit (Cheezum et al., 2001)	4.2	4.0
Centroid (Cheezum et al., 2001)	7.8	6.6
Sum of absolute differences (Cheezum et al., 2001)	6.9	8.1
Cross-correlation (Cheezum et al., 2001)	4.2	6.3

conditions), and a new trajectory starts. This test mimics the situation of finite dilution. The same background and peak values are used as for the previous test (cf. Table 1), but bias and standard deviation are computed on the actual positions (x,y)—rather than the displacements a—as:

$$\text{bias}_x = \langle \tilde{x} - x \rangle; \qquad \text{bias}_y = \langle \tilde{y} - y \rangle, \tag{22}$$

and

$$\sigma_x = \langle (\tilde{x} - x) - \langle (\tilde{x} - x) \rangle)^2 \rangle^{1/2};$$
$$\sigma_y = \langle (\tilde{y} - y) - \langle (\tilde{y} - y) \rangle)^2 \rangle^{1/2}, \tag{23}$$

where $\langle \cdot \rangle$ now denotes the ensemble average over all point detections in a movie. The results are shown in Fig. 6. While the standard deviation is comparable to Fig. 4, the bias values are much lower than in the previous test. This is due to the fact that bias and standard deviation are correlated in the one-dimensional case, whereas they are independent here.

To test the trajectory linking in the case where two particles cross, we consider test movies showing 10 horizontally moving points and 10 vertically moving points, such that each pair of points exactly coincides in a certain frame. The background intensity is again fixed at 10, and the peak intensity of the horizontally moving points is fixed at 23, corresponding to an SNR of 2.71. The peak intensity of the vertically moving points is gradually in-

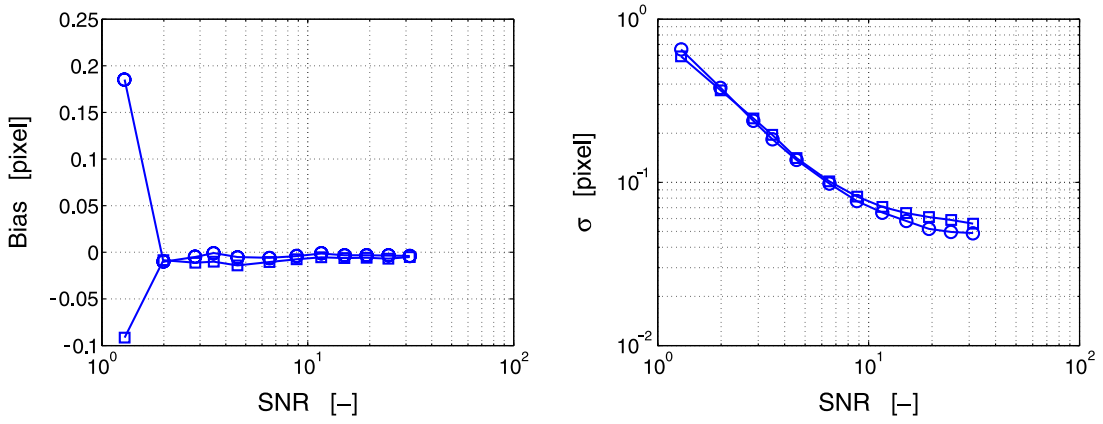

Fig. 6. Left panel: bias versus SNR for 10 point blobs moving at random angular orientations with 0.27 pixel/frame. Right panel: standard deviation versus SNR for the same cases. Each point is averaged from 3500 independent measurements. Circles indicate the x component, squares the y component of the respective measures.

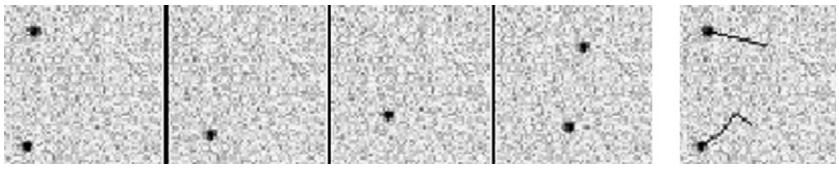

Fig. 7. Sequence of two moving points with the upper one missing in two frames, e.g., due to occlusion or over-sensitive thresholding. The link range is $R = 3$, thus taking three subsequent frames into account for each linking step. The panel to the very right shows the correct recovery of the broken trajectory. (Image intensities are inverted for printing purposes.)

creased. Whenever two particles coincide, only one point observation will be detected. Since the linking algorithm does not allow for a point to be part of several links in any frame, one of the two trajectories must break. In the case where the two point sets are of equal brightness, the choice is random. In 50% of the cases, the vertical trajectory is continuous and the horizontal one pauses, and vice versa for the other 50%. If the point intensities (m_0) however differ, the trajectory of the brighter particle is consistently continued, whereas the dimmer one suffers a gap. This is due to the particular choice of linking cost function, Eq. (14), where differences in m_0 are taken into account, and the fact that the brighter particle "masks" the dimmer one in the local maximum selection. A difference in SNR of 0.15 is sufficient for this to work in 100% of the cases.

The case where a particle temporarily escapes detection is considered in Fig. 7. Extending the link range to $R > 1$ future frames (cf. Section 2.3), successfully prevents gaps in the resulting trajectories, as both points are available for linking.

4. Feature point tracking and trajectory analysis: case studies

Three feature point tracking applications from cell biology are considered:

(1) tracking of endosomes containing fluorescently labeled low-density lipoprotein (DiI-LDL) molecules,
(2) tracking of internalized Adenovirus-2 (Ad-2) particles moving along microtubules, and
(3) tracking of Qdots on the plasma membrane.

These case studies help to demonstrate the robustness and applicability of the algorithm for a wide variety of "feature points." In addition, quantitative analysis based on MSS is introduced to quantify the related dispersive properties.

4.1. Moment scaling spectrum of endosome motion

In the first application, endosomes of 3T6 mouse fibroblast cells are imaged. LDL is fluorescently labeled with DiI red. Endosomes containing DiI-LDL are imaged using TIRFM at 20 Hz with 80 nm/pixel resolution. Two thousand 16-bit TIFF frames are recorded. Fig. 8 shows a few sample frames. The parameters used in tracking are listed in Table 3. The particle is successfully traced over 1446 frames (Fig. 9, top panel) before it fades out.

The analysis of the motion is based on calculating the *moments of displacement*. Let $\mathbf{x}_\ell(n)$ the position vector $(x_\ell(n), y_\ell(n))$ on trajectory ℓ at time $n\Delta t$ for $n = 0, 1, 2, \ldots, M_\ell - 1$ where M_ℓ is the total number of point detections in trajectory ℓ, i.e., its *length*. Δt is the real-time difference between two subsequent frames. The moment of *order* v for a specific *frame shift* Δn, corresponding to a time shift $\delta t = \Delta n \Delta t$, is defined as

$$\mu_{v,\ell}(\Delta n) = \frac{1}{M_\ell - \Delta n} \sum_{n=0}^{M_\ell - \Delta n - 1} |\mathbf{x}_\ell(n + \Delta n) - \mathbf{x}_\ell(n)|^v, \qquad (24)$$

where $|\cdot|$ denotes the 2-norm (Euclidean norm). Missing summands due to non-existing frame numbers ($R > 1$) in the trajectory are skipped and the normalization factor

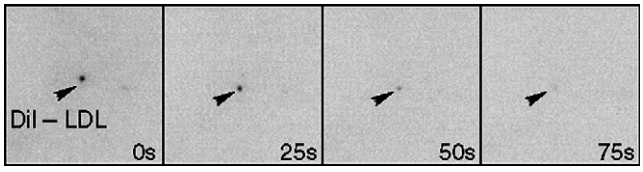

Fig. 8. Time-lapse frame image sequence of a DiI-LDL containing endosome (arrow head) in a 3T6 mouse fibroblast observed using TIRFM and 20 frames/s. Each image shows a 12 μm × 12 μm region on 150 × 150 pixels corresponding to a resolution of 80 nm/pixel. The time difference in seconds to the first image of the sequence is given in the lower-right corner of each frame. Using $R = 1$, the particle is tracked over 1446 frames before fading out. (Image intensities are inverted for printing purposes.)

Table 3
Summary of tracking algorithm parameter settings used in the examples of this section

Parameter	DiI-LDL	Ad-2	Noc	Qdot
Particle radius w [pixel]	4.0	2.0	3.0	3.0
Intensity percentile r [%]	0.1	2.0	1.0	0.05
Cutoff score T_s [–]	0.0	1.0	4.0	0.0
Maximum step length L [pixel]	5.0	5.0	1.0	1.0
Link range R [frames]	1	4	2	1 or 10

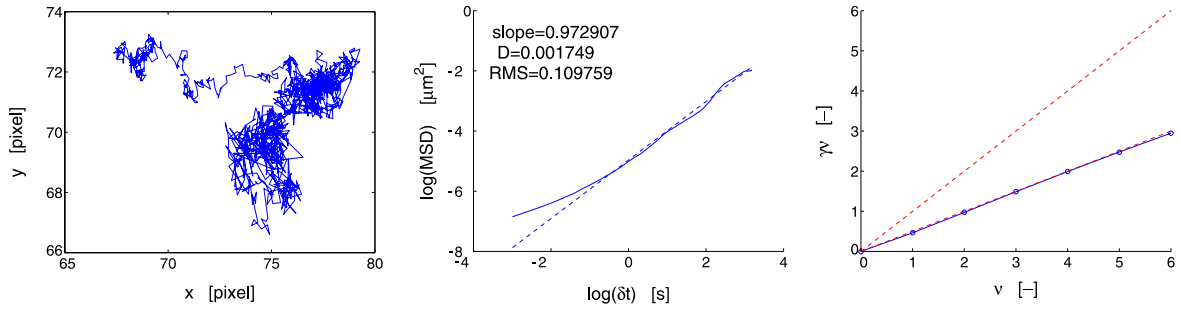

Fig. 9. Tracking an endosome containing fluorescent DiI-LDL. Right panel: *xy*-path of the particle as tracked from the video recording. Middle panel: MSD of the track as defined in Eq. (24). The dashed line is the result of a linear least squares fit to determine the slope and the diffusion constant (values given in the figure). Right panel: MSS of the track.

is accordingly adjusted. The special case of $v = 2$ is called *mean square displacement* (MSD).

To quantify the particle motion (Ferrari et al., 2001), these moments are calculated for $v = 0, 1, 2, \ldots, 6$ and $\Delta n = 1, \ldots, M_\ell/3$, and are drawn versus $\delta t = \Delta n \Delta t$ in a double logarithmic plot (see, e.g., middle panel of Fig. 9). Assuming each moment to depend on the time shift in a power law $\mu_v(\delta t) \propto \delta t^{\gamma_v}$ (Ferrari et al., 2001), all *scaling coefficients* γ_v are determined by a linear least squares regression to $\log\mu_v$ versus $\log\delta t$. In addition, the generalized two-dimensional diffusion coefficients of all orders $v > 0$ are obtained from the y-axis intercepts y_0 as: $D_v = (2v)^{-1} \cdot \exp(y_0)$. D_2 corresponds to the regular diffusion constant in the case of strongly self-similar, pure diffusion. The plot of γ_v versus v is called MSS according to Ferrari et al. (2001). For all strongly self-similar processes, the MSS shows a straight line through the origin as γ_0 is always equal to 0. The slope of this line is an excellent measure for the type of the observed motion. Finding this slope using a linear least squares fit is a very robust procedure due to the almost perfect linearity of the MSS for strongly self-similar processes. Moreover, the MSS slope has good and uniform sensitivity to detect different modes of motion within the same trajectory. For normal (free) and strongly self-similar diffusion, the MSS slope is 1/2. A slope of 1 indicates ballistic, i.e., uniform and directed motion. A slope of 0 characterizes a stationary object. The region between 0 and 1/2 is the sub-diffusive regime (e.g., confined diffusion) and between 1/2 and 1 is the super-diffusive regime (e.g., diffusion with overlayed deterministic drift, Lévy flights). Every strongly self-similar process will yield scaling coefficients γ_v that linearly depend on v. A curved or kinked plot is indicative of a weakly self-similar process (Ferrari et al., 2001).

The results of the MSS analysis for DiI-LDL are shown in the middle and right panels of Fig. 9. The MSS shows an almost perfect straight line of slope 1/2. The particle thus undergoes free and normal diffusion. The diffusion coefficient is determined from the second moment to be $D_2 = 1.7 \times 10^{-3}$ μm²/s. The intensity of the endosome is shown over time in the left panel of Fig. 13. The continuous fading could be due to photobleaching or the endosome moving into the cell and thus out of the TIRF region.

4.2. Tracking and analysis of Adenovirus-2 trafficking

The tracking of microtubule-dependent trafficking of intracellular Ad-2 serves as a test for the algorithm in cases of fast directed motion. We analyze the original 16-bit frame images of Suomalainen et al. (1999) that were tracked by hand for the original publication. Fluorescently labeled internalized Ad-2 particles in wild-type TC7 cells are imaged using a wide-field fluorescence microscope. The resolution is 0.15 μm/pixel and the time interval between frames is 1.3 s. The complete protocol is defined by Suomalainen et al. (1999). Fig. 10 shows a time-lapse sequence of some frames. The unspecific photobleaching is clearly visible. The total movie is 104 frames long. Tracking is done using the parameter values given in Table 3 and yielded 73 tracks of lengths between 60 and 104 frames. Three example tracks are shown in the left panel of Fig. 11 and the intensity of virus particle (a) over all 104 frames is shown in the middle panel of Fig. 13.

The control experiment considers Ad-2 in HeLa cells treated with nocodazole, a microtubule depolymerizing drug. The tracker parameters are given in column

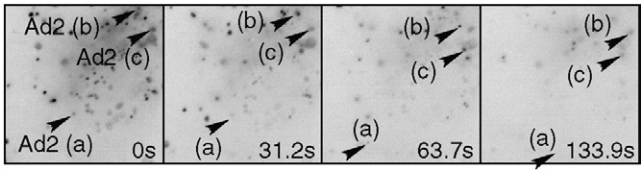

Fig. 10. Time-lapse frame image sequence of Ad-2 moving along microtubules in TC7 cells observed using fluorescence wide-field microscopy and 1.3 s/frame. Each image shows a 37.65 μm × 37.65 μm region on 251 × 251 pixels corresponding to a resolution of 0.15 μm/pixel. The time difference in seconds to the first image of the sequence is given in the lower-right corner of each frame. All indicated particles are tracked over the full 104 frames. (Image intensities are inverted for printing purposes.)

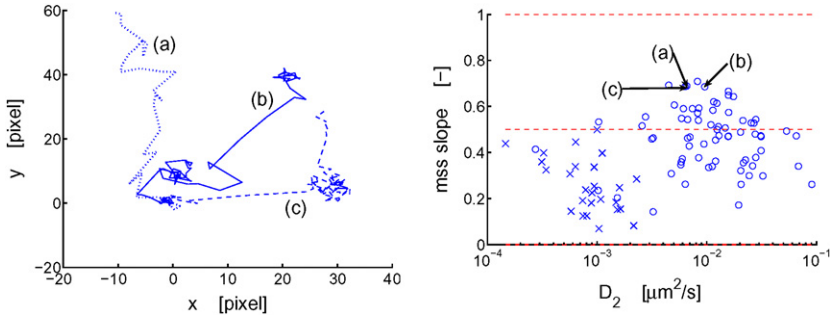

Fig. 11. Trafficking of Ad-2 particles along microtubules. Left panel: *xy*-path of three sample particles as tracked over all 104 frames of the movie. All trajectories are shifted to start at point (0,0). Stretches of directed motion with intermediate random motion are visible. Right panel: scatter plot of overall diffusion constants D_2 and MSS slopes for all tracks of the wild-type experiment (circles) and the nocodazole control (crosses).

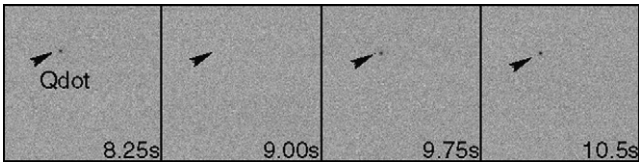

Fig. 12. Time-lapse frame image sequence of a Qdot (arrow head) on the plasma membrane of 3T6 mouse fibroblasts observed using TIRFM and 20 frames/s. Each image shows a 12 μm × 12 μm region on 150 × 150 pixels corresponding to a resolution of 80 nm/pixel. The time difference in seconds to the first image of the sequence is given in the lower-right corner of each frame. Using $R = 1$, the particle can be tracked over 21 frames, with $R = 10$, the longest trajectory spans 1068 frames. (Image intensities are inverted for printing purposes.)

"Noc" of Table 3. The total length of the control movie is 275 frames. Twenty-seven tracks of lengths between 80 and 252 frames are extracted. MSS analysis as outlined in the previous case is done for all recorded Ad-2 tracks. Fig. 11 shows a scatter plot of all diffusion coefficients and MSS slopes for the two experiments. The existence of biased/directed motion in the wild-type experiment is evident from the MSS slope values above 0.5. Still a significant fraction of trajectories with MSS slopes around or below 0.5 exists which means that those particles are not always transported actively. As can be seen from the left panel of the figure, intermediate pauses or changes in direction exist, causing the overall average MSS slope to drop. The nocodazole control never exhibits directed motion and particles are at most freely diffusive, which provides the evidence for the directed motion to depend on microtubules (Suomalainen et al., 1999).

4.3. Tracking of quantum dots

We consider tracking of Qdots to demonstrate the function of the multi-frame linking algorithm as described in Section 2.3 for $R > 1$. Quantum dots (QuantumDot, www.qdots.com) are extremely bright and photostable fluorescent nano-particles. Their signal strength makes them a true alternative to fluorescent proteins. Quantum dots however exhibit strong fluctuations in their emission intensity ("blinking"), which complicates the linking of point detections into trajectories.

Biotinylized ConcanavalinA is bound to 3T6 cells for 30 s in PBS. The cells are dipped in imaging medium and 0.2 μM Streptavidin-coupled 25 nm Qdots are added. Using TIRFM, 2000 frames are recorded at 20 Hz video rate with 80 nm/pixel resolution. The images are stored as uncompressed 16-bit TIFF files.

Fig. 12 shows a few sample frames from the movie. The blinking is clearly visible as the Qdot has vanished

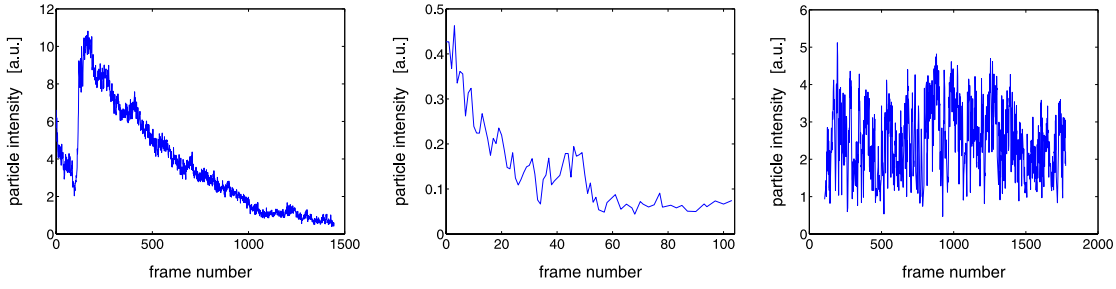

Fig. 13. Particle intensities m_0 over time as returned by the tracking algorithm. The time evolution of the intensity of each of the three test cases is shown. The sum of all pixel values within the particle radius w is computed as intensity measure m_0, cf. Eq. (8). The strong intensity fluctuations ("blinking") of the Qdot can clearly be seen in the right panel. The continuous intensity decay of DiI-LDL and Ad-2 could be due to photobleaching or the particle moving out of focus.

in the second image. Good parameter settings for the tracking algorithm are determined using the graphical user interface. Their values are given in Table 3. Using two subsequent frames to perform trajectory linking (i.e., $R = 1$), the longest track that can be extracted is 21 frames in length. Setting $R = 10$ increases the track length to 1068 frames. This is a clear advantage since tracks as short as those in the $R = 1$ case would not allow to determine diffusion constants or other properties of the motion with significant statistics.

The right panel of Fig. 13 shows the time evolution of the fluorescence intensity of the sample Qdot. The strong fluctuations (''blinking'') are clearly visible, as well as its photostability and brightness. The Qdot in this example is almost stationary. The MSS shows a straight line of slope 0.083 (figure not shown), and the diffusion constant is below the detection limit.

4.4. Experimental tracking quality

To assess the tracking quality, the SNR of the images are estimated using the noise in the bright image regions as outlined earlier. The program used to estimate the SNR is tested on the synthetic images of known SNR from Section 3. The SNR values are correctly determined within $\pm 7\%$. The mean measured SNR of both the DiI-LDL and the Qdot samples is 3.1, averaged over all frames. The background intensity of the Qdot video is more than three times larger than the one of the DiI-LDL case. Using the results from Section 3, this SNR corresponds to both a tracking accuracy and precision of about 0.2 pixel (16 nm). The experimentally measured track standard deviation in the Qdot example is 0.4 pixel, which is consistent with the very small value of its MSS slope and illustrates the sensitivity of the latter measure.

Positioning errors result in observed apparent sub-diffusion (Martin et al., 2002). Using the model of Martin et al. (2002), the measured diffusion coefficient for the DiI-LDL containing endosome, and above estimate of the positioning error, the apparent slope in the double logarithmic MSD plot of DiI-LDL (Fig. 9, middle panel) is predicted to be $0.933 < \gamma_2 < 1$. This is in excellent agreement with our measured γ_2 of 0.973 and supports the conclusion that the motion of the endosome is normal diffusion, as properly indicated by the MSS slope.

5. Conclusions

In this article, we presented a computationally efficient and robust method for two-dimensional feature point tracking that can be used for quantitative time-resolved studies of particle trajectories as they appear in several applications in cell biology. The presented method was demonstrated to be of high accuracy and precision even at moderate SNR, and to provide sub-pixel accuracy in all practical situations. The absence of any intrinsic models regarding the motion of the particles that are being tracked, in combination with its robustness and efficiency, makes the method particularly well suited for biological applications relying on trajectories developed by fluorescence microscopy. If available, prior knowledge about the underlying physical processes can still be incorporated by suitably choosing the cost functional for the trajectory linking.

The presented method emphasizes computational efficiency and ease of use. The former goal is motivated by our observation that many available feature point tracking algorithms suffer from poor computational performance or large memory requirements if long sequences of large images are to be processed (Vallotton et al., 2003). The presented implementation is capable of tracking a sequence of three thousand 214×214 pixel TIFF images in less than 15 s on a 3.06 GHz Intel Pentium 4 desktop computer. Ease of use is achieved by minimizing the number of user-set parameters of the algorithm and providing a user-friendly graphical user interface. The presented implementation only requires the approximate particle radius w, the intensity percentile r, the cutoff score T_s for the non-particle discrimination, the maximum link length L, and the number of future frames for the linker R to be set by the user. Of these five parameters, w and R are trivial. The remaining three have a direct physical meaning and can easily be determined by inspection of a few frames in the movie. The graphical user interface provides additional guidance and support in this process.

The presented case studies have shown that the algorithm performs well on non-smooth (virus particle tracking) as well as smooth (endosome tracking) motions. The Qdot example demonstrated the capabilities of handling blinking objects and creating continuous trajectories from their intermittent detections. The presented MSS analysis is introduced as an appropriate way of analyzing and classifying the recorded trajectories.

The algorithm presented in this paper is not intrinsically limited to two dimensions. Its application to three-dimensional data is straightforward, provided such data are available. The following limitations are however present: in the feature point detection, the algorithm is limited to small (compared to background variations) spherical particles or point spread blobs and the trajectory linking is limited by the specific cost functional one defines. For the cost functional used in this article, the limitation is obviously given by the criterion that two equally bright and equally large particles must always be separated by more than the distance they move per frame. Using different cost functionals this could be relaxed at the expense of other limitations such as a loss of universality due to prior information about the type of motion. Furthermore, we assumed that every detected point corresponds to exactly one particle. The algorithm

is thus unable to resolve particle coalescence or division, or to yield two continuous trajectories if two particles exactly cross in space and time. The non-particle discrimination step is, if used at all, limited by the assumption that the majority of the detected points corresponds to particles of the desired kind.

The presented algorithm is implemented in C as a multi-user multi-tier client–server application. In our experience, this implementation is fast and stable even under high load with several concurrent users. The graphical user interface is implemented in Java and runs on different computer platforms. The source code of both implementations is freely available from the authors under the terms and conditions of the ICoS software release agreement.

Acknowledgments

The authors are grateful to Ingo Oppermann and Remo Marti for their help in implementing the tracking algorithm and the graphical user interface in their final form. The video for the virus tracking case was generously provided by Prof. Urs Greber; both the DiI-LDL as well as the Qdot data were provided by Helge Ewers (Helenius group), and the PTK2 image used to demonstrate the clustering was provided by Dr. Alicia Smith (Helenius group). We are indebted for many enlightening discussions with Prof. Ari Helenius, Dr. Alicia Smith, and Helge Ewers (Institute of Biochemistry, ETH Zürich) that motivated our work.

References

Anderson, C.M., Georgiou, G.N., Morrrison, I.E., Stevenson, G.V., Cherry, R.J., 1992. Tracking of cell surface receptors by fluorescence digital imaging microscopy using a charge-coupled device camera. low-density lipoprotein and influenza virus receptor mobility at 4 °C. J. Cell Sci. 101 (Pt 2), 415–425.

Crocker, J.C., Grier, D.G., 1996. Methods of digital video microscopy for colloidal studies. J. Coll. Interface Sci. 179, 298–310.

Chetverikov, D., Verestóy, J., 1999. Feature point tracking for incomplete trajectories. Computing 62 (4), 321–338.

Cheezum, M.K., Walker, W.F., Guilford, W.H., 2001. Quantitative comparison of algorithms for tracking single fluorescent particles. Biophys. J. 81, 2378–2388.

Dalziel, S.B., 1992. Decay of rotating turbulence: some particle tracking experiments. Appl. Sci. Res. 49, 217–244.

Dalziel, S.B., 1993a. Rayleigh–Taylor instability: experiments with image analysis. Dynam. Atmos. Oceans 20, 127–153.

Dalziel, S.B., 1993b. Decay of rotating turbulence: some particle tracking experiments. In: Nieuwstadt, F.T.M. (Ed.), Flow Visualization And Image Analysis. Kluwer, Dordrecht, pp. 27–54.

DeBrabander, M., Geuens, G., Nuydens, R., Moeremans, M., DeMey, J., 1985. Probing microtubule-dependent intracellular motility with nanometre particle video ultramicroscopy (nanovid ultramicroscopy). Cytobios 43 (174S), 273–283.

Ferrari, R., Manfroi, A.J., Young, W.R., 2001. Strongly and weakly self-similar diffusion. Physica D. 154, 111–137.

Fujiwara, T., Ritchie, K., Murakoshi, H., Jacobson, K., Kusumi, A., 2002. Phospholipids undergo hop diffusion in compartmentalized cell membrane. J. Cell Biol. 157 (6), 1071–1081.

Gelles, J., Schnapp, B.J., Sheetz, M.P., 1998. Tracking kinesin-driven movements with nanometre-scale precision. Nature 331, 450–453.

Ghosh, R.N., Webb, W.W., 1994. Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules. Biophys. J. 66 (5), 1302–1318.

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

Hichcock, F.L., 1941. The distribution of a product from several sources to numerous localities. J. Math. Phys. 20, 224.

Hicks, B.W., Angelides, K.J., 1995. Tracking movements of lipids and Thy1 molecules in the plasmalemma of living fibroblasts by fluorescence video microscopy with nanometer scale precision. J. Membr. Biol. 144 (3), 231–244.

Jain, A.K., 1986. Fundamentals of Image Processing. Prentice-Hall, Englewood Cliffs, NJ.

Lerasle, F., Rives, G., Dhome, M., 1999. Tracking of human limbs by multiocular vision. Comput. Vis. Image Und. 75 (3), 229–246.

Martin, D.S., Forstner, M.B., Käs, J.A., 2002. Apparent subdiffusion inherent to single particle tracking. Biophys. J. 83, 2109–2117.

Morgan, A.D., Dagless, E.L., Milford, D.J., Thomas, B.T., 1990. Road edge tracking for robot road following: a real-time implementation. Image Vision Comput. 8 (3), 233–240.

Pelkmans, L., Kartenbeck, J., Helenius, A., 2001. Caveolar endocytosis of simian virus 40 reveals a new two-step vesicular-transport pathway to the ER. Nat. Cell Biol. 3, 473–483.

Pelkmans, L., Püntener, D., Helenius, A., 2002. Local actin polymerization and dynamin recruitment in SV40-induced internalization of caveolae. Science 296, 535–539.

Qian, H., Sheetz, M.P., Elson, E.L., 1991. Single particle tracking. analysis of diffusion and flow in two-dimensional systems. Biophys. J. 60 (4), 910–921.

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

Sanchiz, J.M., Pla, F., 1999. Feature correspondence and motion recovery in vehicle planar navigation. Pattern Recogn. 32, 1961–1977.

Saxton, M.J., Jacobson, K., 1997. Single-particle tracking: applications to membrane dynamics. Annu. Rev. Biophys. Biomol. Struct. 26, 373–399.

Suomalainen, M., Nakano, M.Y., Keller, S., Boucke, K., Stidwill, R.P., Greber, U.F., 1999. Microtubule-dependent minus and plus end-directed motilities are competing processes for nuclear targeting of adenovirus. J. Cell Biol. 144 (4), 657–672.

Toomre, D., Manstein, D.J., 2001. Lighting up the cell surface with evanescent wave microscopy. Trends Cell Biol. 11 (7), 298–303.

Thompson, R.E., Larson, D.R., Webb, W.W., 2002. Precise nanometer localization analysis for individual fluorescent probes. Biophys. J. 82 (5), 2775–2783.

Vallotton, P., Ponti, A., Waterman-Storer, C.M., Salmon, E.D., Danuser, G., 2003. Recovery, visualization, and analysis of actin and tubulin polymer flow in live cells: a fluorescent speckle microscopy study. Biophys. J. 85, 1289–1306.

Wereley, S.T., Gui, L., Meinhart, C.D., 2002. Advanced algorithms for microscale particle image velocimetry. AIAA J. 40 (6), 1047–1055.

Zhang, F., Crise, B., Su, B., Hou, Y., Rose, J.K., Bothwell, A., Jacobson, K., 1991. Lateral diffusion of membrane-spanning and glycosylphosphatidylinositol-linked proteins: toward establishing rules governing the lateral mobility of membrane proteins. J. Cell Biol. 115 (1), 75–84.