

Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis

It is remarkable that a science which began with the consideration of games of chance should have become the most important object of human knowledge.

— PIERRE-SIMON, MARQUIS DE LAPLACE
Théorie Analytique des Probabilités (1812)

Abstract — Quantitative analysis of dynamic processes in living cells by means of fluorescence microscopy imaging requires tracking of hundreds of bright spots in noisy image sequences. Deterministic approaches, which use object detection prior to tracking, perform poorly in the case of noisy image data. We propose an improved, completely automatic tracker, built within a Bayesian probabilistic framework. It better exploits spatiotemporal information and prior knowledge than common approaches, yielding more robust tracking also in cases of photobleaching and object interaction. The tracking method was evaluated using simulated but realistic image sequences, for which ground truth was available. The results of these experiments show that the method is more accurate and robust than popular tracking methods. In addition, validation experiments were conducted with real fluorescence microscopy image data acquired for microtubule growth analysis. These demonstrate that the method yields results that are in good agreement with manual tracking performed by expert cell biologists. Our findings suggest that the method may replace laborious manual procedures.

Based upon: I. Smal, K. Draegestein, N. Galjart, W. Niessen, E. Meijering, “Particle Filtering for Multiple Object Tracking in Dynamic Fluorescence Microscopy Images: Application to Microtubule Growth Analysis”, *IEEE Transactions on Medical Imaging*, vol. 27, no. 6, pp. 789–804, 2008.

3.1 Introduction

In the past decade, advances in molecular cell biology have triggered the development of highly sophisticated live cell fluorescence microscopy systems capable of *in vivo* multidimensional imaging of subcellular dynamic processes. Analysis of time-lapse image data has redefined the understanding of many biological processes, which in the past had been studied using fixed material. Motion analysis of nanoscale objects such as proteins or vesicles, or subcellular structures such as microtubules (Fig. 3.1), commonly tagged with green fluorescent protein (GFP), requires tracking of large and time-varying numbers of spots in noisy image sequences [54,95,132,135,160,161,166]. Nowadays, high-throughput experiments generate vast amounts of dynamic image data, which cannot be analyzed manually with sufficient speed, accuracy and reproducibility. Consequently, many biologically relevant questions are either left unaddressed, or answered with great uncertainty. Hence, the development of automated tracking methods which replace tedious manual procedures and eliminate the bias and variability in human judgments, is of great importance.

Conventional approaches to tracking in molecular cell biology typically consist of two subsequent steps. In the first step, objects of interest are detected separately in each image frame and their positions are estimated based on, for instance, intensity thresholding [19], multiscale analysis using the wavelet transform [52], or model fitting [161]. The second step solves the correspondence problem between sets of estimated positions. This is usually done in a frame-by-frame fashion, based on nearest-neighbor or smooth-motion criteria [33,171]. Such approaches are applicable to image data showing limited numbers of clearly distinguishable spots against relatively uniform backgrounds, but fail to yield reliable results in the case of poor imaging conditions [26,32]. Tracking methods based on optic flow [13,167] are not suitable because the underlying assumption of brightness preservation over time is not satisfied in fluorescence microscopy, due to photobleaching. Methods based on spatiotemporal segmentation by minimal cost path searching have also been proposed [17,128]. Until present, however, these have been demonstrated to work well only for the tracking of a single object [128], or a very limited number of well-separated objects [17]. As has been observed [17], such methods fail when either the number of objects is larger than a few dozen, or when the object trajectories cross each other, which make them unsuitable for our applications.

As a consequence of the limited performance of existing approaches, tracking is still performed manually in many laboratories worldwide. It has been argued [95] that in order to reach similar superior performance as expert human observers in temporal data association, while at the same time achieving a higher level of sensitivity and accuracy, it is necessary to make better use of temporal information and (application specific) prior knowledge about the morphodynamics of the objects being studied. The human visual system integrates to a high degree spatial, temporal and prior information [23] to resolve ambiguous situations in estimating motion flows in image sequences. Here we explore the power of a Bayesian generalization of the standard Kalman filtering approach in emulating this process. It addresses the problem of estimating the hidden state of a dynamic system by constructing the posterior probability density function (pdf) of the state based on all available information, including

prior knowledge and the (noisy) measurements. Since this pdf embodies all available statistical information, it can be termed a complete solution to the estimation problem.

Bayesian filtering is a conceptual approach, which yields analytical solutions, in closed form, only in the case of linear systems and Gaussian statistics. In the case of non-linearity and non-Gaussian statistics, numerical solutions can be obtained by applying sequential Monte Carlo (SMC) methods [39], in particular particle filtering (PF) [9]. In the filtering process, tracking is performed by using a predefined model of the expected dynamics to predict the object states, and by using the (noisy) measurements (possibly from different types of sensors) to obtain the posterior probability of these states. In the case of multiple target tracking, the main task is to perform efficient measurement-to-target association, on the basis of thresholded measurements [15]. The classical data association methods in multiple target tracking can be divided into two main classes: unique-neighbor data association methods, as in the multiple hypothesis tracker (MHT), which associate each measurement with one of the previously established tracks, and all-neighbors data association methods, such as joint probabilistic data association (JPDA), which use all measurements for updating all track estimates [15]. The tracking performance of these methods is known to be limited by the linearity of the data models. By contrast, SMC methods that propagate the posterior pdf, or methods that propagate the first-order statistical moment (the probability hypothesis density) of the multitarget pdf [90], have been shown to be successful in solving the multiple target tracking and data association problems when the data models are nonlinear and non-Gaussian [68, 104].

Previous applications of PF-based motion estimation include radar- and sonar-based tracking [104, 175], mobile robot localization [39, 184], teleconferencing or video surveillance [115], and other human motion applications [31, 110, 186]. In most computer vision applications, tracking is limited to a few objects only [70, 89]. Most biological applications, on the other hand, require the tracking of large and time-varying numbers of objects. Recently, the use of PF in combination with level-sets [83] and active contours [139] has been reported for biological cell tracking. These methods outperform deterministic methods, but they are straightforward applications of the original algorithm [70] for single target tracking, and cannot be directly applied to the simultaneous tracking of many intracellular objects. A PF-like method for the tracking of proteins has also been suggested [183], but it still uses template matching for the linking stage, it requires manual initialization, and tracks only a single object. In this chapter, we extend our earlier conference reports [143, 144], and develop a fully automated PF-based method for robust and accurate tracking of multiple nanoscale objects in two-dimensional (2D) and three-dimensional (3D) dynamic fluorescence microscopy images. Its performance is demonstrated for a particular biological application of interest: microtubule growth analysis.

The chapter is organized as follows. In Section 3.2 we give more in-depth information on the biological application considered in this chapter, providing further biological motivation for our work. In Section 3.3 we present the general tracking framework and its extension to allow tracking of multiple objects. Next, in Section 3.4, we describe the necessary improvements and adaptations to tailor the framework to the application. These include a new dynamic model which allows dealing with object

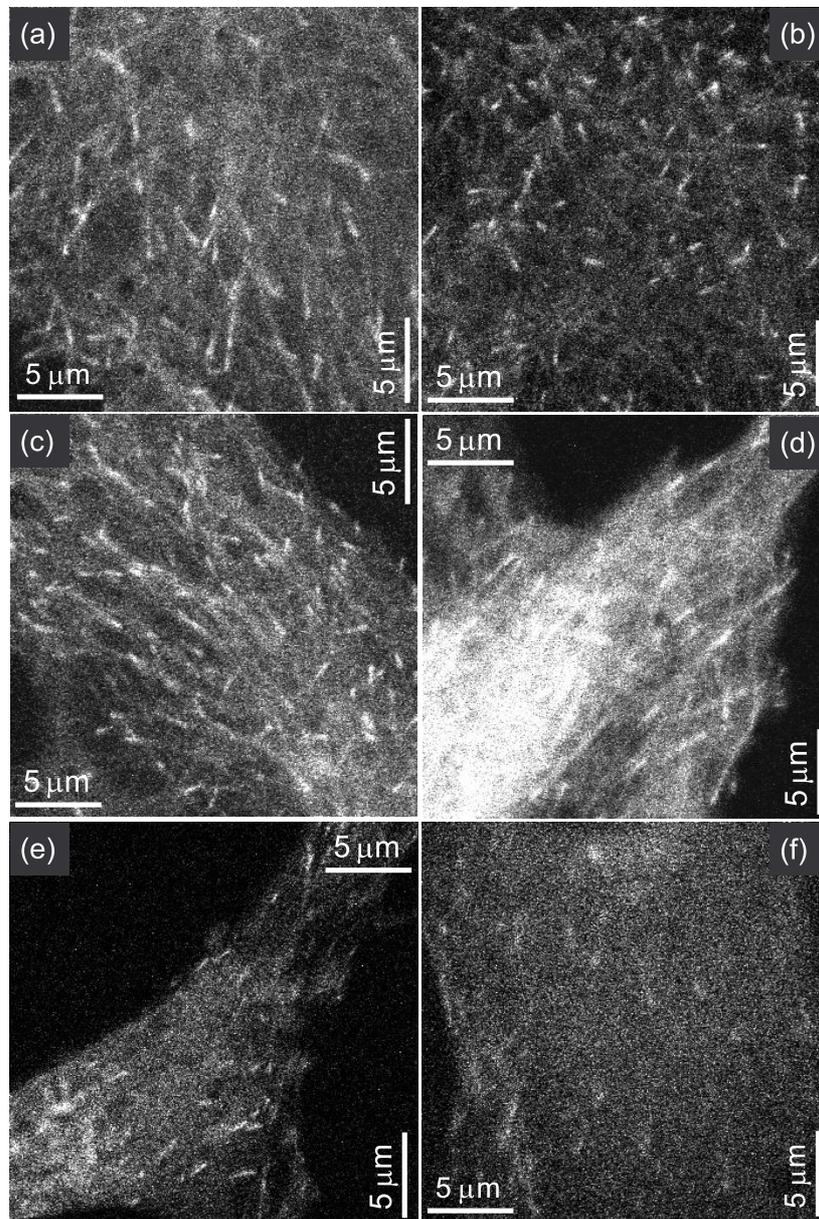


Figure 3.1. Examples of microtubules tagged with GFP-labeled plus end tracking proteins (bright spots), imaged using fluorescence confocal microscopy. The images are single frames from six 2D time-lapse studies, conducted with different experimental and imaging conditions. The quality of such images typically ranges from $\text{SNR} \approx 5\text{--}6$ (a-c) to the extremely low $\text{SNR} \approx 2\text{--}3$ (d-f).

interaction and photobleaching effects. In addition, we improve the robustness and reproducibility of the algorithm by introducing a new importance function for data-dependent sampling (the choice of the importance density is one of the most critical issues in the design of a PF method). We also propose a new, completely automatic track initiation procedure. In Section 3.5, we present experimental results of applying our PF method to synthetic image sequences, for which ground truth was available, as well as to real fluorescence microscopy image data of microtubule growth. A concluding discussion of the main findings and their potential implications is given in Section 3.6.

3.2 Microtubule Growth Analysis

Microtubules (MTs) are polarized tubular filaments (diameter ≈ 25 nm) composed of α/β -tubulin heterodimers. In most cell types, one end of a MT (the minus-end) is embedded in the so-called MT organizing center (MTOC), while the other end (the plus-end) is exposed to the cytoplasm. MT polymerization involves the addition of α/β -tubulin subunits to the plus end. During MT disassembly, these subunits are lost. MTs frequently switch between growth and shrinkage, a feature called dynamic instability [37]. The conversion of growth to shrinkage is called catastrophe, while the switch from shrinkage to growth is called rescue. The dynamic behavior of MTs is described by MT growth and shrinkage rates, and catastrophe and rescue frequencies. MTs are fairly rigid structures having nearly constant velocity while growing or shrinking [48]. MT dynamics is highly regulated, as a properly organized MT network is essential for many cellular processes, including mitosis, cell polarity, transport of vesicles, and the migration and differentiation of cells. For example, when cells enter mitosis, the *cdc2* kinase controls MT dynamics such that the steady-state length of MTs decreases considerably. This is important for spindle formation and positioning [173]. It has been shown that an increase in catastrophe frequency is largely responsible for this change in MT length [172].

Plus-end-tracking proteins, or +TIPs [137], specifically bind to MT plus-ends and have been linked to MT-target interactions and MT dynamics [4, 67, 80]. Plus-end-tracking was first described for overexpressed GFP-CLIP170 in cultured mammalian cells [114]. In time-lapse movies, typical fluorescent “comet-like” dashes were observed, which represented GFP-CLIP170 bound to the ends of growing MTs. As plus-end tracking is intimately associated with MT growth, fluorescently labeled +TIPs are now widely used to measure MT growth rates in living cells, and they are also the objects of interest considered in the present work. With fluorescent +TIPs, all growing MTs can be discerned. Alternatively, the advantage of using fluorescent tubulin is that all parameters of MT dynamics can be measured. However, in regions where the MT network is dense, the fluorescent MT network obscures MT ends, making it very difficult to examine MT dynamics. Hence, in many studies based on fluorescent tubulin [62, 127, 187], analysis is restricted to areas within the cells where the MT network is sparse. Ideally, one should use both methods to acquire all possible knowledge regarding MT dynamics, and this will be addressed in future work.

+TIPs are well positioned to perform their regulatory tasks. A network of inter-

acting proteins, including +TIPs, may govern the changes in MT dynamics that occur during the cell cycle [106]. Since +TIPs are so important and display such a fascinating behavior, the mechanisms by which +TIPs recognize MT ends have attracted much attention. In one view, +TIPs binds to newly synthesized MT ends with high affinity and detach seconds later from the MT lattice, either in a regulated manner or stochastically [114]. However, other mechanisms have also been proposed [4, 27, 67]. Measuring the distribution and displacement of a fluorescent +TIP in time may shed light on the mechanism of MT end binding. However, this is a labor intensive procedure if fluorescent tracks have to be delineated by hand, and very likely leads to user bias and loss of important information. By developing a reliable tracking algorithm we obtain information on the behavior of all growing MTs within a cell, which reveals the spatiotemporal distribution and regulation of growing MTs. Importantly, this information can be linked to the spatiotemporal fluorescent distribution of +TIPs. This is extremely important, since the localization of +TIPs reports on the dynamic state of MTs and the cell.

3.3 Tracking Framework

Before describing the details of our tracking approach, we first recap the basic principles of nonlinear Bayesian tracking in general (Section 3.3.1), and PF in particular (Section 3.3.2), as well as the extension that has been proposed in the literature to allow tracking of multiple objects within this framework (Section 3.3.3).

3.3.1 Nonlinear Bayesian Tracking

The Bayesian tracking approach deals with the problem of inferring knowledge about the unobserved state of a dynamic system, which changes over time, using a sequence of noisy measurements. In a state-space approach to dynamic state estimation, the state vector \mathbf{x}_t of a system contains all relevant information required to describe the system under investigation. Bayesian estimation in this case is used to recursively estimate a time evolving posterior distribution (or filtering distribution) $p(\mathbf{x}_t|\mathbf{z}_{1:t})$, which describes the object state \mathbf{x}_t given all observations $\mathbf{z}_{1:t}$ up to time t .

The exact solution to this problem can be constructed by specifying the Markovian probabilistic model of the state evolution, $D(\mathbf{x}_t|\mathbf{x}_{t-1})$, and the likelihood $L(\mathbf{z}_t|\mathbf{x}_t)$, which relates the noisy measurements to any state. The required probability density function $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ may be obtained, recursively, in two stages: prediction and update. It is assumed that the initial pdf, $p(\mathbf{x}_0|\mathbf{z}_0) \equiv p(\mathbf{x}_0)$, also known as the prior, is available ($\mathbf{z}_{1:0} = \mathbf{z}_0$ being the set of no measurements).

The prediction stage involves using the system model and pdf $p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ to obtain the prior pdf of the state at time t via the Chapman-Kolmogorov equation:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t-1}) = \int D(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})d\mathbf{x}_{t-1}. \quad (3.1)$$

In the update stage, when a measurement \mathbf{z}_t becomes available, Bayes' rule is used to modify the prior density and obtain the required posterior density of the current

state:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \propto L(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}). \quad (3.2)$$

This recursive estimation of the filtering distribution can be processed sequentially rather than as a batch, so that it is not necessary to store the complete data set nor to reprocess existing data if a new measurement becomes available [9]. The filtering distribution embodies all available statistical information and an optimal estimate of the state can theoretically be found with respect to any sensible criterion.

3.3.2 Particle Filtering Methods

The optimal Bayesian solution, defined by the recurrence relations (3.1) and (3.2), is analytically tractable in a restrictive set of cases, including the Kalman filter, which provides an optimal solution in case of linear dynamic systems with Gaussian noise, and grid based filters [9]. For most practical models of interest, SMC methods (also known as bootstrap filtering, particle filtering, and the condensation algorithm [70]) are used as an efficient numerical approximation. The basic idea here is to represent the required posterior density function $p(\mathbf{x}_t | \mathbf{z}_{1:t})$ with a set of N_s random samples, or particles, and associated weights $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_s}$. Thus, the filtering distribution can be approximated as

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \approx \sum_{i=1}^{N_s} w_t^{(i)} \delta(\mathbf{x}_t - \mathbf{x}_t^{(i)}),$$

where $\delta(\cdot)$ is the Dirac delta function and the weights are normalized such that $\sum_{i=1}^{N_s} w_t^{(i)} = 1$. These samples and weights are then propagated through time to give an approximation of the filtering distribution at subsequent time steps.

The weights in this representation are chosen using a sequential version of importance sampling (SIS) [125]. It applies when auxiliary knowledge is available in the form of an importance function $q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t)$ describing which areas of the state-space contain most information about the posterior. The idea is then to sample the particles in those areas of the state-space where the importance function is large and to avoid as much as possible generating samples with low weights, since they provide a negligible contribution to the posterior. Thus, we would like to generate a set of new particles from an appropriately selected proposal function, i.e.,

$$\mathbf{x}_t^{(i)} \sim q(\mathbf{x}_t | \mathbf{x}_{t-1}^{(i)}, \mathbf{z}_t), \quad i = \{1, \dots, N_s\}. \quad (3.3)$$

A detailed formulation of $q(\cdot | \cdot)$ is given in Section 3.4.6.

With the set of state particles obtained from (3.3), the importance weights $w_t^{(i)}$ may be recursively updated as follows:

$$w_t^{(i)} \propto \frac{L(\mathbf{z}_t | \mathbf{x}_t^{(i)}) D(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)})}{q(\mathbf{x}_t^{(i)} | \mathbf{x}_{t-1}^{(i)}, \mathbf{z}_t)} w_{t-1}^{(i)}. \quad (3.4)$$

Generally, any importance function can be chosen, subject to some weak constraints [40, 126]. The only requirements are the possibility to easily draw samples from it

and evaluate the likelihood and dynamic models. For very large numbers of samples, this MC characterization becomes equivalent to the usual functional description of the posterior pdf.

By using this representation, statistical inferences, such as expectation, maximum a posteriori (MAP), and minimum mean square error (MMSE) estimators (the latter is used for the object position estimation in the approach proposed in this chapter), can easily be approximated. For example,

$$\hat{\mathbf{x}}_t^{\text{MMSE}} = \mathbb{E}_p[\mathbf{x}_t] = \int \mathbf{x}_t p(\mathbf{x}_t | \mathbf{z}_{1:t}) d\mathbf{x}_t \approx \sum_{i=1}^{N_s} \mathbf{x}_t^{(i)} w_t^{(i)}. \quad (3.5)$$

A common problem with the SIS particle filter is the degeneracy phenomenon, where after a few iterations, all but a few particles will have negligible weight. The variance of the importance weights can only increase (stochastically) over time [40]. The effect of the degeneracy can be reduced by a good choice of importance density and the use of resampling [9, 40, 125] to eliminate particles that have small weights and concentrate on particles with large weights (see [40] for more details on degeneracy and resampling procedures).

3.3.3 Multi-Modality and Mixture Tracking

It is straightforward to generalize the Bayesian formulation to the problem of multi-object tracking. However, due to the increase in dimensionality, this formulation gives an exponential explosion of computational demands. The primary goal in a multi-object tracking application is to determine the posterior distribution, which is multi-modal in this case, over the current *joint* configuration of the objects at the current time step, given all observations up to that time step. Multiple modes are caused either by ambiguity about the object state due to insufficient measurements, which is supposed to be resolved during tracking, or by measurements coming from multiple objects being tracked. Generally, MC methods are poor at consistently maintaining the multi-modality in the filtering distribution. In practice it frequently occurs that all the particles quickly migrate to one of the modes, subsequently discarding other modes.

To capture and maintain the multi-modal nature, which is inherent to many applications in which tracking of multiple objects is required, the filtering distribution is explicitly represented by an M -component mixture model [174]:

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) = \sum_{m=1}^M \pi_{m,t} p_m(\mathbf{x}_t | \mathbf{z}_{1:t}), \quad (3.6)$$

with $\sum_{m=1}^M \pi_{m,t} = 1$ and a non-parametric model is assumed for the individual mixture components. In this case, the particle representation of the filtering distribution, $\{\mathbf{x}_t^{(i)}, w_t^{(i)}\}_{i=1}^N$ with $N = MN_s$ particles, is augmented with a set of component indicators, $\{c_t^{(i)}\}_{i=1}^N$, with $c_t^{(i)} = m$ if particle i belongs to mixture component m . For the mixture component m we also use the equivalent notation $\{\mathbf{x}_{m,t}^{(l)}, w_{m,t}^{(l)}\}_{l=1}^{N_s} =$

$\{\mathbf{x}_t^{(i)}, w_t^{(i)} : c_t^{(i)} = m\}_{i=1}^N$. The representation (3.6) can be updated in the same fashion as the two-step approach for standard Bayesian sequential estimation [174].

3.4 Tailoring the Framework

Having presented the general framework for PF-based multiple object tracking, we now tailor it to our application: the study of MT dynamics. This requires making choices regarding the models involved as well as a number of computational and practical issues. Specifically, we propose a new dynamic model, which does not only cover spatiotemporal behavior but also allows dealing with photobleaching effects (Section 3.4.1) and object interaction (Section 3.4.2). In addition, we propose a new observation model and corresponding likelihood function (Section 3.4.3), tailored to objects that are elongated in their direction of motion. The robustness and computational efficiency of the algorithm are improved by using two-step hierarchical searching (Section 3.4.4), measurement gating (Section 3.4.5) and a new importance function for data-dependent sampling (Section 3.4.6). Finally, we propose practical procedures for particle reclustering (Section 3.4.7) and automatic track initiation (Section 3.4.8).

3.4.1 State-Space and Dynamic Model

In order to model the dynamic behavior of the visible ends of MTs in our algorithm, we represent the object state with the state vector $\mathbf{x}_t = (x_t, \dot{x}_t, y_t, \dot{y}_t, z_t, \dot{z}_t, \sigma_{\max,t}, \sigma_{\min,t}, \sigma_{z,t}, I_t)^T$, where $(\sigma_{\max,t}, \sigma_{\min,t}, \sigma_{z,t})^T \triangleq \mathbf{s}_t$ is the object shape feature vector (see Section 3.4.3), $(x_t, y_t, z_t)^T \triangleq \mathbf{r}_t$ is the radius vector, $\dot{\mathbf{r}}_t \triangleq \mathbf{v}_t$ is velocity, and I_t object intensity. The state evolution model $D(\mathbf{x}_t|\mathbf{x}_{t-1})$ can be factorized as

$$D(\mathbf{x}_t|\mathbf{x}_{t-1}) = D_y(\mathbf{y}_t|\mathbf{y}_{t-1})D_s(\mathbf{s}_t|\mathbf{s}_{t-1})D_I(I_t|I_{t-1}), \quad (3.7)$$

where $\mathbf{y}_t = (x_t, \dot{x}_t, y_t, \dot{y}_t, z_t, \dot{z}_t)$. Here, $D_y(\mathbf{y}_t|\mathbf{y}_{t-1})$ is modeled using a linear Gaussian model [40], which can easily be evaluated pointwise in (3.4), and is given by

$$D_y(\mathbf{y}_t|\mathbf{y}_{t-1}) \propto \exp\left(-\frac{1}{2}(\mathbf{y}_t - \mathbf{F}\mathbf{y}_{t-1})^T \mathbf{Q}^{-1}(\mathbf{y}_t - \mathbf{F}\mathbf{y}_{t-1})\right), \quad (3.8)$$

with the process transition matrix $\mathbf{F} = \text{diag}[\mathbf{F}_1, \mathbf{F}_1, \mathbf{F}_1]$ and covariance matrix $\mathbf{Q} = \text{diag}[\mathbf{Q}_1, \mathbf{Q}_1, \mathbf{Q}_1]$ given by

$$\mathbf{F}_1 = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{Q}_1 = \begin{pmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{pmatrix},$$

where T is the sampling interval. Depending on the parameters q_{11} , q_{12} , q_{22} the model (3.8) describes a variety of motion patterns, ranging from random walk ($\|\mathbf{v}_t\| = 0$, $q_{11} \neq 0$, $q_{12} = 0$, $q_{22} = 0$) to nearly constant velocity ($\|\mathbf{v}_t\| \neq 0$, $q_{11} \neq 0$, $q_{12} \neq 0$, $q_{22} \neq 0$) [11], [84]. In our application, the parameters are fixed to $q_{11} = \frac{q_1}{3}T^3$, $q_{12} = \frac{q_1}{2}T^2$, $q_{22} = q_1T$, where q_1 controls the noise level. In this case, model (3.8) corresponds to the continuous-time model $\dot{\mathbf{r}}(t) = \mathbf{w}(t) \approx 0$, where $\mathbf{w}(t)$ is white

noise that corresponds to noisy accelerations [11]. We also make the realistic assumption that object velocities are bounded. This prior information is object dependent and will be used for state initialization (see Section 3.4.8). Small changes in frame-to-frame MT appearance (shape) are modeled using the Gaussian transition prior $D_s(\mathbf{s}_t|\mathbf{s}_{t-1}) = \mathcal{N}(\mathbf{s}_t|\mathbf{s}_{t-1}, Tq_2\mathbf{I})$, where $\mathcal{N}(\cdot|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ indicates the normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, \mathbf{I} is the identity matrix, and q_2 represents the noise level in object appearance.

In practice, the analysis of time-lapse fluorescence microscopy images is complicated by photobleaching, a dynamic process by which the fluorescent proteins undergo photoinduced chemical destruction upon exposure to excitation light and thus lose their ability to fluoresce. Although the mechanisms of photobleaching are not yet well understood, two commonly used (and practically similar) approximations of fluorescence intensity over time are given by

$$I(t) = Ae^{-at} + B \quad (3.9)$$

and

$$I(t) = I_0 \left(1 + \left(\frac{t}{L} \right)^k \right)^{-1}, \quad (3.10)$$

where A , B , a , I_0 , L , and k are experimentally determined constants (see [124, 148] for more details on the validity and sensitivity of these models). The rate of photobleaching is a function of the excitation intensity. With a laser as an excitation source, photobleaching is observed on the time scale of microseconds to seconds. The high numerical aperture objectives currently in use, which maximize spatial resolution and improve the limits of detection, further accelerate the photobleaching process. Commonly, photobleaching is ignored by standard tracking methods, but in many practical cases it is necessary to model this process so as to be less sensitive to changing experimental conditions.

Following the common approximation (3.9), we model object intensity in our image data by the sum of a time-dependent, a time-independent, and a random component:

$$I_t + I_c + u_t = \frac{I_0\hat{A}}{\hat{A} + \hat{B}} e^{-\hat{\alpha}t} + \frac{I_0\hat{B}}{\hat{A} + \hat{B}} + u_t, \quad (3.11)$$

where u_t is zero-mean Gaussian process noise and I_0 is the initial object intensity, obtained by the initialization procedure (see Section 3.4.8). The parameters \hat{A} , \hat{B} , and $\hat{\alpha}$ are estimated using the Levenberg-Marquardt algorithm for nonlinear fitting of (3.9) to the average background intensity over time, b_t (see Section 3.4.3). In order to conveniently incorporate the photobleaching effect contained in (3.11) into our framework, we approximate it as a first-order Gauss-Markov process, $I_t = (1 - \hat{\alpha})I_{t-1} + u_t$, which models the exponential intensity decay in the discrete-time domain. In this case, the corresponding state prior $D_I(I_t|I_{t-1}) = \mathcal{N}(I_t|(1 - \hat{\alpha})I_{t-1}, q_3T)$, where $q_3 = T^{-1}\sigma_u^2$ and σ_u^2 is the variance of u_t .

The photobleaching effect could alternatively be accommodated in our framework by assuming a constant intensity model ($\hat{\alpha} = 0$) for $D_I(I_t|I_{t-1})$, but with a very high variance for the process noise, σ_u^2 . However, in practice, because of the limited number

of MC samples, the variance of the estimation would rapidly grow, and many samples would be used inefficiently, causing problems especially in the case of a highly peaked likelihood $L(\mathbf{z}_t|\mathbf{x}_t)$ (see Section 3.4.3). By using (3.11), we follow at least the trend of the intensity changes, and bring the estimation closer to the optimal solution. This way, we reduce the estimation variance and, consequently, the number of MC samples needed for the same accuracy as in the case of the constant intensity model.

In summary, the proposed model (3.7) correctly approximates small accelerations in object motion and fluctuations in object intensity, and therefore is very suitable for tracking growing MTs, as their dynamics can be well modeled by constant velocity plus small random diffusion [48]. The model (3.8) can also be successfully used for tracking other subcellular structures, for example vesicles, which are characterized by motion with higher nonlinearity. In that case, the process noise level, defined by \mathbf{Q} , should be increased.

3.4.2 Object Interactions and Markov Random Field

In order to obtain a more realistic motion model and avoid track coalescence in the case of multiple object tracking, we explicitly model the interaction between objects using a Markov random field (MRF) [76]. Here we use a pairwise MRF, expressed by means of a Gibbs distribution

$$\begin{aligned} \psi_t(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}) &\propto \exp(-d_t^{i,j}), \\ i, j &\in \{1, \dots, N\}, \quad c_t^{(i)} \neq c_t^{(j)}, \end{aligned} \quad (3.12)$$

where $d_t^{i,j}$ is a penalty function which penalizes the states of two objects $c_t^{(i)}$ and $c_t^{(j)}$ that are closely spaced at time t . That is, $d_t^{i,j}$ is maximal when two objects coincide and gradually falls off as they move apart. This simple pairwise representation is easy to implement yet can be made quite sophisticated. Using this form, we can still retain the predictive motion model of each individual target. To this end, we sample N_s times the pairs $(\mathbf{x}_{m,t-1}^{(l)}, \mathbf{x}_{m,t}^{(l)})$ (M such pairs at a time, $m = \{1, \dots, M\}$), from $p_m(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1})$ and $q(\mathbf{x}_t|\mathbf{x}_{m,t-1}^{(l)}, \mathbf{z}_t)$, respectively, $l = \{1, \dots, N_s\}$. Taking into account (3.12), the weights (3.4) in this case are given by

$$w_{m,t}^{(l)} \propto \frac{L(\mathbf{z}_t|\mathbf{x}_{m,t}^{(l)})D(\mathbf{x}_{m,t}^{(l)}|\mathbf{x}_{m,t-1}^{(l)})}{q(\mathbf{x}_{m,t}^{(l)}|\mathbf{x}_{m,t-1}^{(l)}, \mathbf{z}_t)} \prod_{k=1, k \neq m}^M \psi_t(\mathbf{x}_{m,t}^{(l)}, \mathbf{x}_{k,t}^{(l)}). \quad (3.13)$$

The mixture representation $\{\{\mathbf{x}_{m,t}^{(l)}, w_{m,t}^{(l)}\}_{m=1}^M\}_{l=1}^{N_s}$ is then straightforwardly transformed to $\{\mathbf{x}_t^{(i)}, w_t^{(i)}, c_t^{(i)}\}_{i=1}^N$. In our application we have found that an interaction potential based only on object positions is sufficient to avoid most tracking failures. The use of a MRF approach is especially relevant and efficient in the case of 3D+t data analysis, because object merging is not possible in our application.

3.4.3 Observation Model and Likelihood

The measurements in our application are represented by a sequence of 2D or 3D images showing the motion of fluorescent proteins. The individual images (also called

frames) are recorded at discrete instants t , with a sampling interval T , with each image consisting of $N_x \times N_y \times N_z$ pixels ($N_z = 1$ in 2D). At each pixel (i, j, k) , which corresponds to a rectangular volume of dimensions $\Delta_x \times \Delta_y \times \Delta_z$ nm³, the measured intensity is denoted as $z_t(i, j, k)$. The complete measurement recorded at time t is an $N_x \times N_y \times N_z$ matrix denoted as $\mathbf{z}_t = \{z_t(i, j, k) : i = 0, \dots, N_x - 1, j = 0, \dots, N_y - 1, k = 0, \dots, N_z - 1\}$. For simplicity we assume that the origins and axis orientations of the (x, y, z) reference system and the (i, j, k) system coincide. Let $\tilde{z}_t(\mathbf{r})$ denote a first-order interpolation of $z_t(\Delta_x i, \Delta_y j, \Delta_z k)$.

The image formation process in a microscope can be modeled as a convolution of the true light distribution coming from the specimen, with a point-spread function (PSF), which is the output of the optical system for an input point light source. The theoretical diffraction-limited PSF in the case of paraxial and non-paraxial imaging can be expressed by the scalar Debye diffraction integral [190]. In practice, however, a 3D Gaussian approximation of the PSF [161] is commonly favored over the more complicated PSF models (such as the Gibson-Lanni model [55]). This choice is mainly motivated by computational considerations, but a Gaussian approximation of the physical PSF is fairly accurate for reasonably large pinhole sizes (relative squared error (RSE) $< 9\%$) and nearly perfect for typical pinhole sizes (RSE $< 1\%$) [190]. In most microscopes currently used, the PSF limits the spatial resolution to ≈ 200 nm in-plane and ≈ 600 nm in the direction of the optical axis, as a consequence of which subcellular structures (typically of size < 20 nm) are imaged as blurred spots. We adopt the common assumption that all blurring processes are due to a linear and spatially invariant PSF.

The PF framework accommodates any PSF that can be calculated pointwise. To model the imaged intensity profile of the object with some shape, one would have to use the convolution with the PSF for every state $\mathbf{x}_t^{(i)}$. In order to overcome this computational overload, we propose to model the PSF and object shape at the same time using the 3D Gaussian approximation. To model the manifest elongation in the intensity profile of MTs, we utilize the velocity components from the state vector \mathbf{x}_t as parameters in the PSF. In this case, for an object of intensity I_t at position \mathbf{r}_t , the intensity contribution to pixel (i, j, k) is approximated as

$$\begin{aligned} h_t(i, j, k; \mathbf{x}_t) &= b_t + (I_t + I_c) \times \\ &\exp\left(-\frac{1}{2} \mathbf{m}^T \mathbf{R}^T \boldsymbol{\Sigma}^{-1} \mathbf{R} \mathbf{m}\right) \times \\ &\exp\left(-\frac{(k\Delta_z - z_t \|\mathbf{m}\| \tan \theta)^2}{2\sigma_z^2}\right), \end{aligned} \quad (3.14)$$

where b_t is the background intensity, σ_z (≈ 235 nm) models the axial blurring, $\mathbf{R} = \mathbf{R}(\phi)$ is a rotation matrix

$$\begin{aligned} \mathbf{R}(\phi) &= \begin{pmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_m^2(\theta) & 0 \\ 0 & \sigma_{\min}^2 \end{pmatrix}, \\ \mathbf{m} &= \begin{pmatrix} i\Delta_x - x_t \\ j\Delta_y - y_t \end{pmatrix}, \quad \sigma_m(\theta) = \sigma_{\min} - (\sigma_{\min} - \sigma_{\max}) \cos \theta, \end{aligned}$$

$$\tan \theta = \frac{\dot{z}_t}{\sqrt{\dot{x}_t^2 + \dot{y}_t^2}}, \quad \tan \phi = \frac{\dot{y}_t}{\dot{x}_t}, \quad -\pi < \phi, \theta \leq \pi.$$

The parameters σ_{\max} and σ_{\min} represent the amount of blurring and, at the same time, model the elongation of the object along the direction of motion. For subresolution structures such as vesicles, $\sigma_{\min} = \sigma_{\max} \approx 80$ nm, and for the elongated MTs $\sigma_{\min} \approx 100$ nm and $\sigma_{\max} \approx 300$ nm.

For background level estimation we use the fact that the contribution of object intensity values to the total image intensity (mainly formed by background structures with lower intensity) is negligible, especially in the case of low SNRs. We have found that in a typical 2D image of size $10^3 \times 10^3$ pixels containing a thousand objects, the number of object pixels is only about 1%. Even if the object intensities would be 10 times as large as the background level (very high SNR), their contribution to the total image intensity would be less than 10%. In that case, the normalized histogram of the image \mathbf{z}_t can be approximated by a Gaussian distribution with mean \hat{b} and variance σ_b^2 . The estimated background $b_t = \hat{b}$ is then calculated according to

$$b_t = \frac{1}{N_x N_y N_z} \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} z_t(i, j, k). \quad (3.15)$$

In the case of a skewed histogram of image intensity, the median of the distribution can be taken as an estimate of the background level. The latter is preferable because it treats object pixels as outliers for the background distribution.

Since an object will affect only the pixels in the vicinity of its location, \mathbf{r}_t , we define the likelihood function as

$$L_G(\mathbf{z}_t | \mathbf{x}_t) \triangleq \prod_{(i,j,k) \in C(\mathbf{x}_t)} \frac{p_h(z_t(i, j, k) | \mathbf{x}_t)}{p_b(z_t(i, j, k) | b_t)}, \quad (3.16)$$

where $C(\mathbf{x}_t) = \{(i, j, k) \in \mathbb{Z}^3 : h_t(i, j, k; \mathbf{x}_t) - b_t > 0.1I_t\}$,

$$p_h(z_t(i, j, k) | \mathbf{x}_t) \propto \frac{1}{\sigma_h(i, j, k)} \exp\left(-\frac{(z_t(i, j, k) - h_t(i, j, k; \mathbf{x}_t))^2}{2\sigma_h^2(i, j, k)}\right), \quad (3.17)$$

and

$$p_b(z_t(i, j, k) | b_t) \propto \exp\left(-\frac{(z_t(i, j, k) - b_t)^2}{2\sigma_b^2}\right), \quad (3.18)$$

with $\sigma_h^2(i, j, k)$ and σ_b^2 the variances of the measurement noise for the object + background and background, respectively, which are assumed to be independent from pixel to pixel and from frame to frame. Poisson noise, which can be used to model the effect of the quantum nature of light on the measured data, is one of the main sources of noise in fluorescence microscopy imaging. The recursive Bayesian solution is applicable as long as the statistics of the measurement noise is known for each pixel. In this chapter we use a valid approximation of Poisson noise, with $\sigma_h^2(i, j, k) = h_t(i, j, k; \mathbf{x}_t)$ and $\sigma_b^2 = b_t$, by scaling the image intensities in order to satisfy the condition $\sigma_b^2 = b_t$ [26].

3.4.4 Hierarchical Searching

Generally, the likelihood $L_G(\mathbf{z}_t|\mathbf{x}_t)$ is very peaked (even when the region $C(\mathbf{x}_t)$ is small) and may lead to severe sample impoverishment and divergence of the filter. Theoretically it is impossible to avoid the degeneracy phenomenon, where, after a few iterations of the algorithm, all but one of the normalized importance weights are very close to zero [40]. Consequently, the accuracy of the estimator also degrades enormously [125]. A commonly used measure of degeneracy is the *estimated effective sample size* [40], given by

$$N_{eff}(t) = \left(\sum_{i=1}^{N_s} (w_t^{(i)})^2 \right)^{-1}, \quad (3.19)$$

which intuitively corresponds to the number of “useful” particles. Degeneracy is usually strong for image data with low SNR, but the filter also performs poorly when the noise level is too small [39]. This suggests that MC estimation with accurate sensors may perform worse than with inaccurate sensors. The problem can be partially fixed by using an observation model which overestimates the measurement noise. While the performance is better, this is not a principled way of fixing the problem; the observation model is artificially inaccurate and the resulting estimation is no longer a posterior, even if infinitely many samples were used. Other methods that try to improve the performance of PF include partitioned sampling [89], the auxiliary particle filter (APF) [9], [126] and the regularized particle filters (RPF) [39, 126]. Because of the highly nonlinear observation model and dynamic model with a high noise level, the mentioned methods are inefficient for our application. Partitioned sampling requires the possibility to partition the state space and to decouple the observation model for each of the partitions, which cannot be done for our application. Application of the APF is beneficial only when the dynamic model is correctly specified with a small amount of process noise. The tracking of highly dynamic structures with linear models requires increasing the process noise in order to capture the typical motion patterns.

To overcome these problems, we use a different approach, based on RPF, and mainly on progressive correction [39]. First, we propose a second observation model:

$$L_S(\mathbf{z}_t|\mathbf{x}_t) \triangleq \frac{\sigma_B}{\sigma_S(\mathbf{x}_t)} \exp \left(\frac{(S_t^z(\mathbf{x}_t) - S_t^b(\mathbf{x}_t))^2}{2\sigma_B^2} - \frac{(S_t^z(\mathbf{x}_t) - S_t^h(\mathbf{x}_t))^2}{2\sigma_S^2(\mathbf{x}_t)} \right), \quad (3.20)$$

where

$$S_t^z(\mathbf{x}_t) = \sum_{(i,j,k) \in C(\mathbf{x}_t)} z_t(i, j, k),$$

and

$$S_t^h(\mathbf{x}_t) = \sum_{(i,j,k) \in C(\mathbf{x}_t)} h_t(i, j, k; \mathbf{x}_t),$$

$S_t^b = b_t |C(\mathbf{x}_t)|$, where $|\cdot|$ denotes the set size operator, and the variances σ_S^2 and σ_B^2 are taken to approximate the Poisson distribution: $\sigma_S^2 = S_t^o$ and $\sigma_B^2 = S_t^b$. The likelihood $L_S(\mathbf{z}_t|\mathbf{x}_t)$ is less peaked but gives an error of the same order as $L_G(\mathbf{z}_t|\mathbf{x}_t)$. Another advantage is that $L_S(\mathbf{z}_t|\mathbf{x}_t)$ can be used for objects without a predefined

shape; only the region $C(\mathbf{x}_t)$, which presumably contains the object, and the total object intensity in $C(\mathbf{x}_t)$ need to be specified.

Subsequently, we propose a modified hierarchical search strategy, which uses both models, L_S and L_G . To this end, we calculate an intermediate state at time t' , between time points $t-1$ and t , by propagating and updating the samples using the likelihood L_S according to

$$\bar{p}(\mathbf{x}_{t'}|\mathbf{z}_{1:t'}) \propto L_S(\mathbf{z}_{t'}|\mathbf{x}_{t'})D(\mathbf{x}_{t'}|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{z}_{1:t-1}) \quad (3.21)$$

where $\mathbf{z}_{t'} = \mathbf{z}_t$. After this step, N_{eff} is still rather high, because the likelihood L_S is less peaked than L_G . In a next step, particles with high weights at time t' are diversified and put into regions where the likelihood L_G is high, giving a much better approximation of the posterior:

$$p(\mathbf{x}_t|\mathbf{z}_{1:t}) \propto L_G(\mathbf{z}_t|\mathbf{x}_t)\mathcal{N}(\mathbf{x}_t|\boldsymbol{\mu}_{t'}, \boldsymbol{\Sigma}_{t'})\bar{p}(\mathbf{x}_{t'}|\mathbf{z}_{1:t'}), \quad (3.22)$$

where the expectation and the variance are given by

$$\boldsymbol{\mu}_{t'} = \mathbb{E}_{\bar{p}}[\mathbf{x}_{t'}], \quad \boldsymbol{\Sigma}_{t'} = \mathbb{E}_{\bar{p}}[(\mathbf{x}_{t'} - \boldsymbol{\mu}_{t'}) (\mathbf{x}_{t'} - \boldsymbol{\mu}_{t'})^T]. \quad (3.23)$$

The described hierarchical search strategy is further denoted as L_{SG} . It keeps the number N_{eff} quite large and, in practice, provides filters that are more stable in time, with lower variance in the position estimation.

3.4.5 Measurement Gating

Multiple object tracking requires gating, or measurement selection. The purpose of gating is to reduce computational expense by eliminating measurements which are far from the predicted measurement location. Gating is performed for each track at each time step t by defining a subvolume of the image space, called the gate. All measurements positioned within the gate are selected and used for the track update step, (3.2), while measurements outside the gate are ignored in these computations. In standard approaches to tracking, using the Kalman filter or extended Kalman filter, measurement gating is accomplished by using the predicted measurement covariance for each object and then updating the predicted state using joint probabilistic data association [79]. In the PF approach, which is able to cope with nonlinear and non-Gaussian models, the analog of the predicted measurement covariance is not available and can be constructed only by taking, for example, a Gaussian approximation of the current particle cloud and using it to perform gating. Generally, this approximation is unsatisfactory, since the advantages gained from having a representation of a non-Gaussian pdf are lost. In the proposed framework, however, this approximation is justified by using the highly peaked likelihood functions and the reclustering procedure (described in Section 3.4.7), which keep the mixture components unimodal.

Having the measurements $\tilde{z}_t(\mathbf{r}_t)$, we define the gate for each of the tracks as follows:

$$C_{m,t} = \{\mathbf{r}_t \in \mathbb{R}^3 : (\mathbf{r}_t - \bar{\mathbf{r}}_{m,t})^T \boldsymbol{\Sigma}_{m,t}^{-1} (\mathbf{r}_t - \bar{\mathbf{r}}_{m,t}) \leq C_0\}, \quad (3.24)$$

where the parameter C_0 specifies the size of the gate, which is proportional to the probability that the object falls within the gate. Generally, since the volume of the

gate is dependent on the tracking accuracy, it varies from scan to scan and from track to track. In our experiments, $C_0 = 9$ (a 3-standard-deviation level gate). The gate $C_{m,t}$ is centered at the position predicted from the particle representation of $p_m(\mathbf{x}_t|\mathbf{z}_{1:t-1})$:

$$\bar{\mathbf{r}}_{m,t} = \mathbb{E}_{p_m}[\mathbf{r}_t] = \int \mathbf{r}_t p_m(\mathbf{x}_t|\mathbf{z}_{1:t-1}) d\mathbf{x}_t \approx \sum_{\substack{i=1, \\ c_{t-1}^{(i)}=m}}^N \bar{\mathbf{r}}_t^{(i)} w_{t-1}^{(i)}, \quad (3.25)$$

where the $\bar{\mathbf{r}}_t^{(i)}$ are the position elements of the state vector

$$\bar{\mathbf{x}}_t^{(i)} \sim D(\mathbf{x}_t|\mathbf{x}_{t-1}^{(i)}), \quad i = \{1, \dots, N\}.$$

Similarly, the covariance matrix is calculated as

$$\boldsymbol{\Sigma}_{m,t} = \mathbb{E}_{p_m}[(\mathbf{r}_t - \bar{\mathbf{r}}_{m,t})(\mathbf{r}_t - \bar{\mathbf{r}}_{m,t})^T]. \quad (3.26)$$

3.4.6 Data-Dependent Sampling

Basic particle filters [9, 70, 143], which use the proposal distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) = D(\mathbf{x}_t|\mathbf{x}_{t-1})$ usually perform poorly because too few samples are generated in regions where the desired posterior $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ is large. In order to construct a proposal distribution which alleviates this problem and takes into account the most recent measurements \mathbf{z}_t , we propose to transform the image sequence into probability distributions. True spots are characterized by a combination of convex intensity distributions and a relatively high intensity. Noise-induced local maxima typically exhibit a random distribution of intensity changes in all directions, leading to a low local curvature [161]. These two discriminative features (intensity and curvature) are used to construct an approximation of the likelihood $L(\mathbf{z}_t|\mathbf{x}_t)$, using the image data available at time t . For each object we use the transformation

$$\tilde{p}_m(\mathbf{r}_t|\mathbf{z}_t) = \frac{(G_\sigma * \tilde{z}_t(\mathbf{r}_t) - b_t)^r \kappa_t^s(\mathbf{r}_t)}{\int_{C_{m,t}} (G_\sigma * \tilde{z}_t(\mathbf{r}_t) - b_t)^r \kappa_t^s(\mathbf{r}_t) dx dy dz}, \quad (3.27)$$

$\forall \mathbf{r}_t \in C_{m,t}$, where G_σ is the Gaussian kernel with standard deviation (scale) σ , the curvature $\kappa_t(\mathbf{r}_t)$ is given by the determinant of the Hessian matrix \mathbf{H} of the intensity $\tilde{z}_t(\mathbf{r}_t)$:

$$\kappa_t(\mathbf{r}_t) = \det(\mathbf{H}(\mathbf{r}_t)), \quad \mathbf{H}(\mathbf{r}_t) = \nabla \cdot \nabla^T \tilde{z}_t(\mathbf{r}_t), \quad (3.28)$$

and the exponents $r > 0$ and $s > 0$ weigh each of the features and determine the peakedness of the likelihood.

Using this transformation, we define the new data dependent proposal distribution for object m as

$$\tilde{q}_m(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) = \tilde{p}_m(\mathbf{r}_t|\mathbf{z}_t) \mathcal{N}(I_t|\tilde{z}_t(\mathbf{r}_t) - b_t, q_3 T) \times \mathcal{N}(\mathbf{s}_t|\mathbf{s}_{m,t-1}^{\text{MMSE}}, T q_2 \mathbf{I}) \mathcal{N}(\mathbf{v}_t|\mathbf{r}_t - \hat{\mathbf{r}}_{m,t-1}^{\text{MMSE}}, T q_1 \mathbf{I}), \quad (3.29)$$

Contrary to the original proposal distribution, which fails if the likelihood is too peaked, the distribution (3.29) generates samples that are highly consistent with the most recent measurements in the predicted (using the information from the previous time step) gates. A combination of both proposal distributions gives excellent results:

$$q_m(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) = \gamma D(\mathbf{x}_t|\mathbf{x}_{t-1}) + (1 - \gamma)\tilde{q}_m(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t),$$

where $0 < \gamma < 1$. Comparison shows that the proposal distribution $q_m(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t)$ is uniformly superior to the regular one ($\gamma = 1$) and scales much better to smaller sample sizes.

3.4.7 Clustering and Track Management

The representation of the filtering distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ as the mixture model (3.6) allows for a deterministic spatial reclustering procedure $(\{c'_t\}, M') = F(\{\mathbf{x}_t\}, \{c_t\}, M)$ [174]. The function F can be implemented in any convenient way. It calculates a new mixture representation (with possibly a different number of mixture components) taking as input the current mixture representation. This allows modeling and capturing merging and splitting events, which also have a direct analogy with biological phenomena. In our implementation, at each iteration the mixture representation is recalculated by applying K -means clustering algorithm. The reclustering is based on spatial information (object positions) only and is initialized with the estimates (3.25).

Taking into account our application, two objects are not allowed to merge when their states become similar. Whenever objects pass close to one another, the object with the best likelihood score typically “hijacks” the particles of the nearby mixture components. As mentioned above, this problem is partly solved by using the MRF model for object interactions. The MRF model significantly improves the tracking performance in 3D+t. For 2D+t data sets, however, the observed motion is a projection of the real 3D motion onto the 2D plane. In this case, when one object passes above or beneath another (in 3D), we perceive the motion as penetration or merging. These situations are in principle ambiguous and frequently cannot be resolved uniquely, neither by an automatic tracking method nor by a human observer.

We detect possible object intersections during tracking by checking whether the gates $C_{m,t}$ intersect each other. For example, for two trajectories, the intersection is captured if $C_{i,t} \cap C_{j,t} \neq \emptyset$, $i, j \in \{1, \dots, M\}$. In general, the measurement space $C_t = \cup_{m=1}^M C_{m,t}$ is partitioned into a set of disjoint regions $C_t = \{C_{1,t}^*, \dots, C_{K,t}^*\}$, where $C_{k,t}^*$ is either the union of connected gates or the gate itself. For each $C_{k,t}^*$, we define a set of indices $J_{k,t}$, which indicate which of the gates $C_{i,t}$ belong to it:

$$J_{k,t} = \{i \in \{1, \dots, M\} : C_{i,t} \in C_{k,t}^*\} \quad (3.30)$$

For the gates $C_{k,t}^*$ with $|J_{k,t}| = 1$, the update of the MC weights $w_{m,t}^{(i)}$ is done according to (3.4). For all other gates $C_{k,t}^*$, which correspond to object interaction, we follow the procedure similar to the one described in Section 3.4.2. For each $C_{k,t}^*$ for which $|J_{k,t}| \neq 1$, the set of states $\{\mathbf{x}_{j,t}^{(l)}\}$, $j \in J_{k,t}$, is sampled from the proposal distribution (for every $l = \{1, \dots, N_s\}$), and a set of hypotheses $\Theta_{k,t}^{(l)} = \{\theta_1^{(l)}, \dots, \theta_S^{(l)}\}$, $S = 2^{|J_{k,t}|}$,

is formed. Each $\theta_i^{(l)}$ is a set of binary associations, $\{a_{i,j}^{(l)}\}$, $j \in J_{k,t}$, where $a_{i,j}^{(l)} = 1$ if object j exists during the interaction, and $a_{i,j}^{(l)} = 0$ if the object “dies” or leaves just before or during the interaction and gives no measurements at time t . The hypothesis that maximizes the likelihood is selected as

$$\hat{\theta}_k^{(l)} = \operatorname{argmax}_{\theta_i^{(l)} \in \Theta_{k,t}^{(l)}} L(\mathbf{z}_t | \mathbf{x}_t), \quad (3.31)$$

where the likelihood $L(\mathbf{z}_t | \mathbf{x}_t)$ can be either $L_G(\mathbf{z}_t | \mathbf{x}_t)$ or $L_S(\mathbf{z}_t | \mathbf{x}_t)$, but the region $C(\mathbf{x}_t)$ is defined as $C(\mathbf{x}_t) = \cup_{j \in J_{k,t}} C(\mathbf{x}_{j,t}^{(l)})$, and $h_t(\cdot; \mathbf{x}_t)$ is substituted in (3.16) and (3.20) for each $\theta_i^{(l)}$ with $\sum_{j \in J_{k,t}} a_{i,j}^{(l)} h_t(\cdot; \mathbf{x}_{j,t}^{(l)})$. For the update of the MC weights $w_{j,t}^{(l)}$ the region $C(\mathbf{x}_t) = C(\mathbf{x}_{j,t}^{(l)})$ and $h_t(\cdot; \mathbf{x}_t) = \sum_{j \in J_{k,t}} \hat{a}_j^{(l)} h_t(\cdot; \mathbf{x}_{j,t}^{(l)})$ are used in (3.16) and (3.20), with the $\hat{a}_j^{(l)}$ denoting the $a_{i,j}^{(l)}$ corresponding to $\hat{\theta}_k^{(l)}$. Additionally, in such cases, we do not perform reclustering, but keep the labels for the current iteration as they were before. If the component representation in the next few frames after the interaction event becomes too diffuse, and there is more than one significant mode, splitting is performed and a new track is initiated (see Section 3.4.8 for more details).

Finally, for the termination of an existing track, the methods commonly used for small target tracking [68, 104] cannot be applied straightforwardly. These methods assume that, due to imperfect sensors, the probability of detecting an object is less than one, and they try to follow the object after disappearance for 4-5 frames, predicting its position in time and hoping to catch it again. In our case, when the density of objects in the images is high, such monitoring would definitely result in “confirming” measurements after 3-5 frames of prediction, but these measurements would very likely originate from another object. In our algorithm in order to terminate the track we define the thresholds $\bar{\sigma}_{\max}$, $\bar{\sigma}_{\min}$, $\bar{\sigma}_z$ that describe the “biggest” objects that we are going to track. Then we sample the particles in the predicted gates $C_{m,t}$ using the data-dependent sampling (3.27) with $s = 0$. If the determinant of the covariance matrix computed for those MC samples is greater than $\bar{\sigma}_{\max}^2 \bar{\sigma}_{\min}^2 \bar{\sigma}_z^2 r^{-3}$ the track is terminated. If the gate $C_{m,t}$ does not contain a real object the determinant value will be much higher than the proposed threshold, which is nicely separate the objects from the background structures.

3.4.8 Initialization and Track Initiation

The prior distribution $p(\mathbf{x}_0)$ is specified based on information available in the first frame. One way to initialize the state vector \mathbf{x}_0 would be to point on the desired bright spots in the image or to select regions of interest. In the latter case, the state vector is initialized by a uniform distribution over the state space, in predefined intervals for velocity and intensity, and the expected number of objects should be specified. During filtering and reclustering, after a burn-off period of 2-3 frames, only the true objects will remain.

For completely automatic initiation of object tracks in the first frame, and also for the detection of potential objects for tracking in subsequent frames, we use the following procedure. First, the image space is divided into $N_I = N_X \times N_Y \times N_Z$

rectangular 3D cells of dimensions $\Delta_c \times \Delta_c \times \Delta_a$, with $\Delta_c = 6\sigma_{\max}$ and $\Delta_a = 6\sigma_z$. Next, for each time step t , the image is converted to a probability map according to (3.27), and $N = MN_s$ particles $\tilde{\mathbf{x}}_t^{(i)}$ are sampled with equal weights. The number of particles in each cell represents the degree of belief in object birth. To discriminate potential objects from background structures or noise, we estimate for each cell the center of mass $\hat{\mathbf{r}}_k$ ($k = \{1, \dots, N_I\}$) by MC integration over that cell and calculate the number of MC samples $n_{k,t}$ in the ellipsoidal regions $S_{k,t}(\mathbf{r}_t)$ centered at $\hat{\mathbf{r}}_k$ (with semi-axes of lengths $\Delta_c/2, \Delta_c/2, \Delta_a/2$). In order to initiate a new object, two conditions have to be satisfied. The first condition is that $n_{k,t}$ should be greater than $N \frac{|S_{k,t}|}{|\mathbf{z}_t|} = N\pi(6N_I)^{-1}$. The threshold represents the expected number of particles if the sampling was done from the image region with uniform background intensity. The second condition is similar to the one for track termination (see Section 3.4.7): the determinant of the covariance matrix should be smaller than $\bar{\sigma}_{\max}^2 \bar{\sigma}_{\min}^2 \bar{\sigma}_z^2 r^{-3}$.

Each object d (out of M_d newly detected at time t) is initialized with mixture weight $\pi_{d,t} = (M + M_d)^{-1}$ and object position $\mathbf{r}_{d,t}$ (the center of mass calculated by MC integration over the region $S_{d,t}(\mathbf{r}_t)$). The velocity is uniformly distributed in a predefined range and the intensity is obtained from the image data for that frame and position. In cases where the samples from an undetected object are split between four cells (in the unlikely event when the object is positioned exactly on the intersection of the cell borders), the object will most probably be detected in the next time frame.

3.5 Experimental Results

The performance of the described PF-based tracking method was evaluated using both computer generated image data (Section 3.5.1) and real fluorescence microscopy image data from MT dynamics studies (Section 3.5.2). The former allowed us to test the accuracy and robustness to noise and object interaction of our algorithm compared to two other commonly used tracking tools. The experiments on real data enabled us to compare our algorithm to expert human observers.

3.5.1 Evaluation on Synthetic Data

3.5.1.1 Simulation Setup

The algorithm was evaluated using synthetic but realistic 2D image sequences (20 time frames of 512×512 pixels, $\Delta_x = \Delta_y = 50$ nm, $T = 1$ sec) of moving MT-like objects (a fixed number of 10, 20, or 40 objects per sequence, yielding data sets of different object densities), generated according to (3.8) and (3.14), for different levels of Poisson noise (see Fig. 3.2) in the range SNR=2–7, since SNR=4 has been identified by previous studies [26, 32] as a critical level at which several popular tracking methods break down. In addition, the algorithm was tested using 3D synthetic image sequences (20 time frames of 512×512 pixels \times 20 optical slices, $\Delta_x = \Delta_y = 50$ nm, $\Delta_z = 200$ nm, $T = 1$ sec, with 10–40 objects per sequence), also for different noise levels in the range of SNR=2–7. Here, SNR is defined as the difference in intensity between the object and the background, divided by the standard deviation of the object noise [32]. The

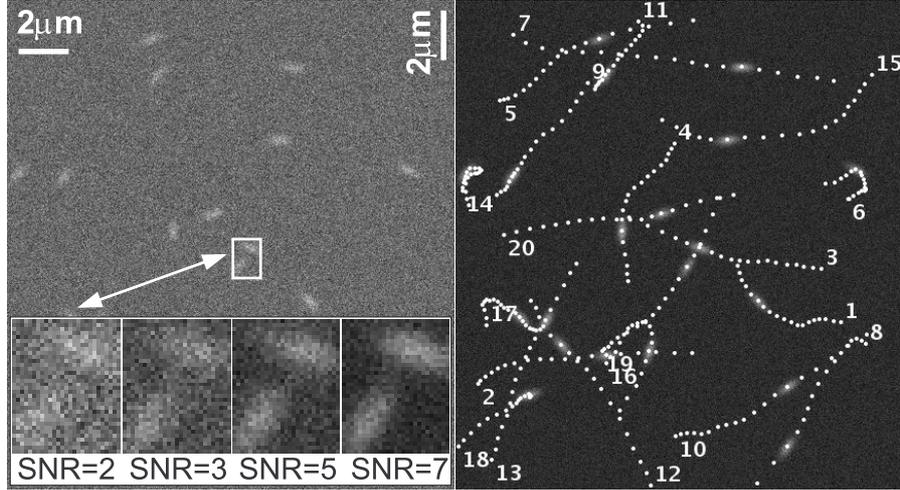


Figure 3.2. Examples of synthetic images used in the experiments. The left image is a single frame from one of the sequences, at SNR=2, giving an impression of object appearance. The insets show zooms of objects at different SNRs. The right image is a frame from another sequence, at SNR=7, with the trajectories of the 20 moving objects superimposed (white dots), illustrating the motion patterns allowed by the linear state evolution model (3.8).

velocities of the objects ranged from 200 to 700 nm/sec, representative of published data [155].

Having the ground truth for the synthetic data, we evaluated the accuracy of tracking by using a traditional quantitative performance measure: the root mean square error (RMSE), in K independent runs (we used $K = 3$) [104]:

$$\text{RMSE} = \sqrt{\frac{1}{K} \sum_{i=1}^K \text{RMSE}_k^2}, \quad (3.32)$$

with

$$\text{RMSE}_k^2 = \frac{1}{M} \sum_{m=1}^M \left\{ \frac{1}{|\mathcal{T}_m|} \sum_{t \in \mathcal{T}_m} \|\mathbf{r}_{m,t} - \hat{\mathbf{r}}_{m,t}^k\|^2 \right\}, \quad (3.33)$$

where $\mathbf{r}_{m,t}$ defines the true position of object m at time t , $\hat{\mathbf{r}}_{m,t}^k$ is a posterior mean estimate of $\mathbf{r}_{m,t}$ for the k th run, and \mathcal{T}_m is the set of time points at which object m exists.

3.5.1.2 Experiments with Hierarchical Searching

In order to show the advantage of using the proposed hierarchical search strategy (see Section 3.4.4), we calculated the localization error at different SNRs for objects moving along horizontal straight lines at a constant speed of 400 nm/sec (similar

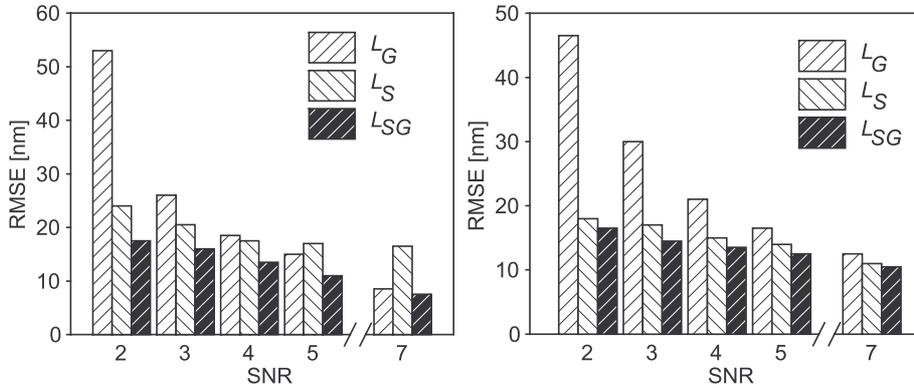


Figure 3.3. The RMSE in object position estimation as a function of SNR for round (left) and elongated (right) objects using the three different observation models, L_G , L_S , and L_{SG} .

to [132]). The tracking was done for two types of objects: round ($\sigma_{\max} = \sigma_{\min} = 100$ nm) and elongated ($\sigma_{\max} = 300$ nm, $\sigma_{\min} = 100$ nm) using the likelihoods L_S , L_G , and the combined two-step approach L_{SG} . The filtering was performed with 500 MC samples. The RMSE for all three models is shown in Fig. 3.3. The localization error of the hierarchical search is lower and the effective sample size N_{eff} is higher than in the case of using only L_G . For comparison, for the likelihoods L_S , L_G , and L_{SG} , the ratios between the effective sample size N_{eff} and N_s are less than 0.5, 0.005, and 0.05, respectively.

3.5.1.3 Comparison with Conventional Two-Stage Tracking Methods

The proposed PF-based tracking method was compared to conventional two-stage (completely separated detection and linking) tracking approaches commonly found in the literature. To maximize the credibility of these experiments, we chose to use two existing, state-of-the-art multitarget tracking software tools based on this principle, rather than making our own (possibly biased) implementation of described methods. The first is *Volocity* (Improvision, Coventry, UK), which is a commercial software package, and the second is *ParticleTracker* [132], which is freely available as a plugin to the public-domain image analysis tool ImageJ [121] (National Institutes of Health, Bethesda, MD, USA).

With *Volocity*, the user has to specify thresholds for the object intensity and the approximate object size in order to discriminate objects from the background, in the detection stage. These thresholds are set globally, for the entire image sequence. Following the extraction of all objects in each frame, linking is performed on the basis of finding nearest neighbors in subsequent image frames. This association of nearest neighbors also takes into account whether the motion is smooth or erratic. With *ParticleTracker*, the detection part also requires setting intensity and

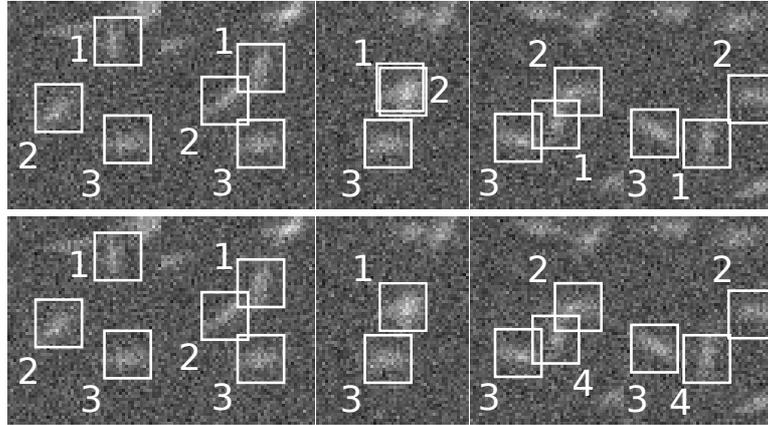


Figure 3.4. Example (SNR=3) showing the ability of our PF method to deal with one-frame occlusion scenarios (top sequence), using the proposed reclustering procedure, while *ParticleTracker* (and similarly *Velocity*) fails (bottom sequence).

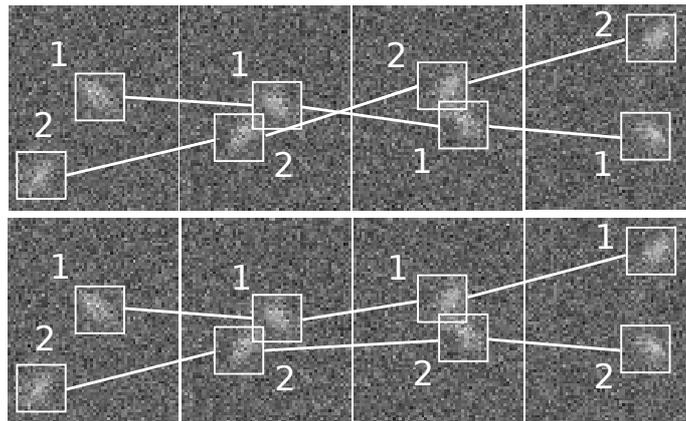


Figure 3.5. Typical example (SNR=3) showing the ability of our PF method to resolve object crossing correctly (top sequence), by using the information about the object shape during the measurement-to-track association process, while *ParticleTracker* (and similarly *Velocity*) fails (bottom sequence).

object size thresholds. The linking, however, is based on finding the global optimal solution for the correspondence problem in a given number of successive frames. The solution is obtained using graph theory and global energy minimization [132]. The linking also utilizes the zeroth- and second-order intensity moments of the object intensities. This better resolves intersection problems and improves the linking result. For both tools, the parameters were optimized manually during each stage, until all objects in the scene were detected. Our PF-based method was initialized using the automatic initialization procedure described in Section 3.4.8. The user-definable algorithm parameters were fixed to the following values: $\sigma_{\max} = 250$ nm, $\sigma_{\min} = 120$ nm,

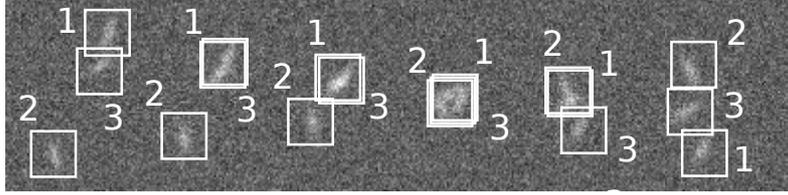


Figure 3.6. Example (SNR=3) where our PF method as well as *ParticleTracker* and *Volocity* failed (only the true tracks are shown in the sequence), because three objects interact at one location and the occlusion lasts for more than one frame.

Table 3.1. Comparison of the ability of the three methods to track objects correctly in cases of object appearance, disappearance, and interactions.

SNR	Volocity		ParticleTracker		Particle Filter	
	r_0	r_1	r_0	r_1	r_0	r_1
	$N_{tr} = 10$					
2	1.1	0.9	1.8	0.1	1	1
3	1	1	1	0.5	1	1
4	1	1	1	0.7	1	1
5	1	1	1	1	1	1
7	1	1	1	1	1	1
	$N_{tr} = 20$					
2	1.15	0.5	2	0.1	1.05	0.8
3	1.05	0.6	1.95	0.15	1	0.9
4	1.05	0.6	1.35	0.45	1	0.95
5	1	0.7	1.1	0.65	1	1
7	1	0.85	1.05	0.9	1	1
	$N_{tr} = 40$					
2	1.9	0.05	1.7	0.1	1.05	0.5
3	1.1	0.6	1.5	0.15	1.02	0.7
4	1.05	0.7	1.42	0.2	1	0.8
5	1.04	0.8	1.22	0.35	1	0.9
7	1.02	0.8	1.17	0.33	1	0.9

$q_1 = 7500 \text{ nm}^2/\text{sec}^3$, $q_2 = 25 \text{ nm}/\text{sec}$, $q_3 = 0.1$, and 10^3 MC samples were used per object. To enable comparisons with manual tracking, five independent, expert observers also tracked the 2D synthetic image sequences, using the freely available software tool MTrackJ [94].

3.5.1.4 Tracking Results

First, using the 2D synthetic image sequences, we compared the ability of our algorithm, *Volocity*, and *ParticleTracker* to track objects correctly, despite possible

object appearances, disappearances, and interactions or crossings. The results of this comparison are presented in Table 3.1. Two performance measures are listed: r_0 , which is the ratio between the number of tracks produced by the algorithm and the true number of tracks present in the data (N_{tr}), and r_1 , which is the ratio between the number of correctly detected tracks and the true number of tracks. Ideally, the values for both ratios should be equal to 1. A value of $r_0 > 1$ indicates that the method produced broken tracks. The main cause of this is the inability to resolve track intersections in some cases (see Fig. 3.4 for an example). In such situations the method either initiates new tracks after the object interaction event (because during the detection stage only one object was detected at that location, see Fig. 3.4), increasing the ratio r_0 , or it incorrectly interchanges the tracks before and after the interaction (see Fig. 3.5 for an example), lowering the ratio r_1 . From the results in Table 3.1 and the examples in Figs. 3.4 and 3.5, it clearly follows that our PF method is much more robust in dealing with object interactions. The scenario in the latter example causes no problems for the PF, as, contrary to two other methods, it exploits information about object appearance. During the measurement-to-track association, the PF favors measurements that are close to the predicted location and that have an elongation in the predicted direction of motion. In some cases (see Fig. 3.6 for an example), all three methods fail, which generally occurs when the interaction is too complicated to resolve even for expert biologists.

Using the same data sets and tracking results, we calculated the RMSE in object position estimation, as a function of SNR. To make a fair comparison, only the results of correctly detected tracks were included in these calculations. The results are shown in Fig. 3.7. The localization error of our algorithm is in the range of 10–50 nm, depending on the SNR, which is approximately 2–3 times smaller than for manual tracking. The error bars represent the interobserver variability for manual tracking, which, together with the average errors, indicate that the performance of manual tracking degrades significantly for low SNRs, as expected. The errors of the three automated methods show the same trend, with our method being consistently more accurate than the other two. This may be explained by the fact that, in addition to object localization by center-of-mass estimation, our hierarchical search performs further localization refinement during the second step (3.22). The RMSE in Fig. 3.7 is larger than in Fig. 3.3, because, even though only correct tracks were included, the accuracy of object localization during multiple object tracking is unfavorably influenced at places where object interaction occurs.

Our algorithm was also tested on the 3D synthetic image sequences as described, using 20 MC simulations. The RMSEs for the observation model L_{SG} ranged from ≈ 30 nm (SNR = 7) to ≈ 70 nm (SNR = 2). These errors were comparable to the errors produced by *Volocity* (in this test, *ParticleTracker* was excluded, as it is limited to tracking in 2D+t). Despite the fact that the axial resolution of the imaging system is approximately three times lower, the localization error was not affected dramatically relative to the 2D+t case. The reason for this is that in 3D+t data, we have a larger number of informative image elements (voxels). As a result, the difference in the RMSEs produced by the estimators employed in our algorithm and in *Volocity* is less compared to Fig. 3.7.

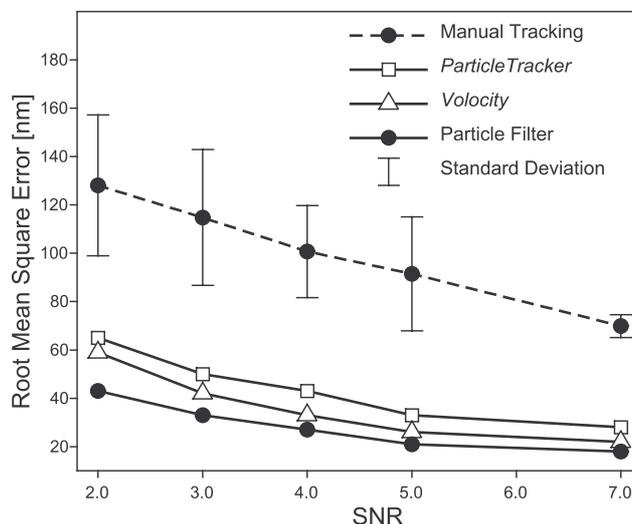


Figure 3.7. The RMSE in object position estimation as a function of SNR for our algorithm (Particle Filter) versus the two other automatic methods (*Volocity* and *ParticleTracker*) and manual tracking (five observers) based on synthetic image data.

3.5.2 Evaluation on Real Data

3.5.2.1 Image Acquisition

In addition to the computer generated image data, real 2D fluorescence microscopy image sequences of MT dynamics were acquired. COS-1 cells were cultured and transfected with GFP-tagged proteins as described [5,155]. Cells were analyzed at 37°C on a Zeiss 510 confocal laser scanning microscope (LSM-510). In most experiments the optical slice separation (in the z -dimension) was set to 1 μm . Images of GFP+TIP movements in transfected cells were acquired every 1–3.5 seconds. For different imaging setups, the pixel size ranged from $70 \times 70 \text{ nm}^2$ to $110 \times 110 \text{ nm}^2$. Image sequences of 30–50 frames were recorded and movies assembled using LSM-510 software. Six representative data sets (30 frames of size 512×512 pixels), examples of which are shown in Fig. 3.1, were preselected from larger volumes by manually choosing the regions of interest. GFP+TIP dashes were tracked in different cell areas. Instantaneous velocities of dashes were calculated simply by dividing measured or tracked distances between frames by the temporal sampling interval.

3.5.2.2 Comparison with Manual Tracking

Lacking ground truth for the real data, we evaluated the performance of our algorithm by visual comparison with manual tracking results. In this case, the latter were obtained from two expert cell biologists, each of which tracked 10 moving MTs of

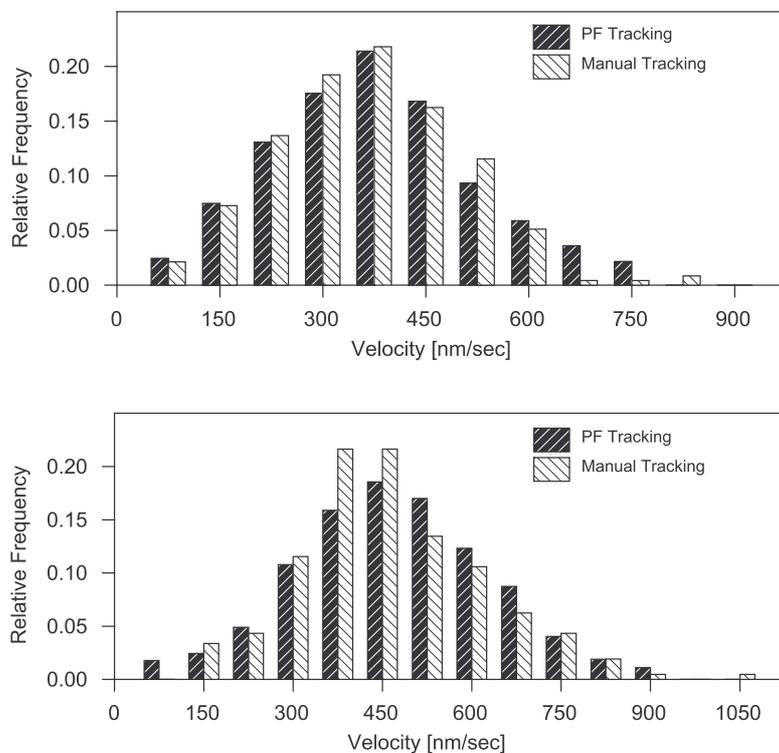


Figure 3.8. Examples of velocity distributions obtained with our automatic tracking algorithm versus manual tracking applied to real fluorescence microscopy image sequences of growing MTs. Results are shown for the data sets in Fig. 3.1(a) (top) and Fig. 3.1(f) (bottom).

interest by using the aforementioned software tool MTrackJ. The selection of target MTs to be tracked was made independently by the two observers. Also, the decision of which feature to track (the tip, the center, or the brightest point) was left to the observers. When done consistently, this does not influence velocity estimations, which is what we focused on in these experiments. The parameters of our algorithm (run with the model L_{SG}) were fixed to the same values as in the case of the evaluation on synthetic data.

3.5.2.3 Tracking Results

Distributions of instant velocities estimated using our algorithm versus manual tracking are presented in Fig. 3.8. The graphs show the results for the data sets of Fig. 3.1(a) and (f), for which $\text{SNR} \approx 5$ and $\text{SNR} \approx 2$, respectively. A visual comparison of the estimated velocities per track, for each of the 10 tracks (the average track length was 13 time steps), is presented in Fig. 3.9, with more details for two

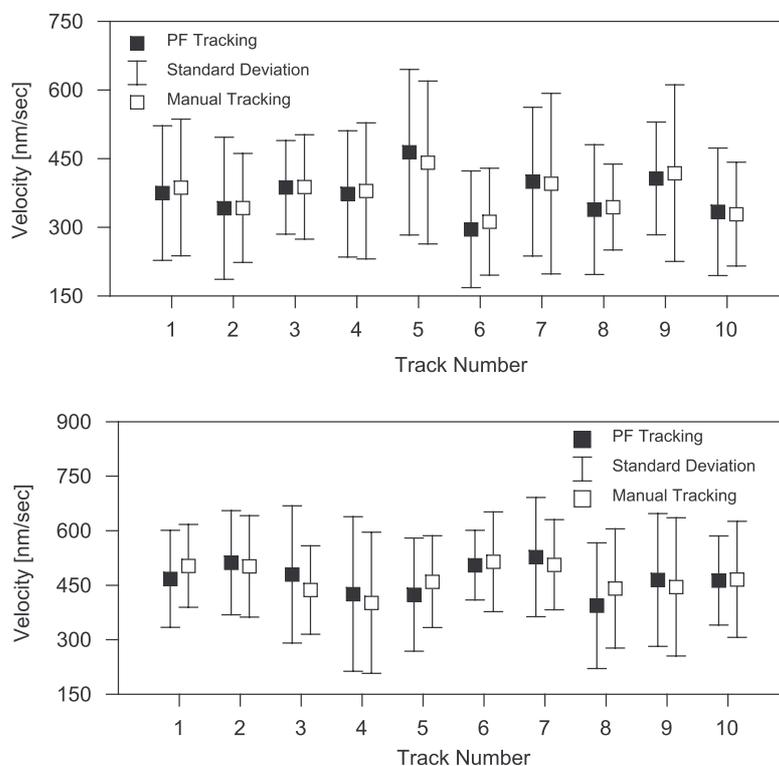


Figure 3.9. Results of velocity estimation for 10 representative MT objects in real fluorescence microscopy image sequences using our automatic tracking algorithm versus manual tracking for the data sets in Fig. 3.1(a) (top) and Fig. 3.1(f) (bottom). Shown are the mean values (black or white squares) and ± 1 standard deviation (bars) of the estimates.

representative tracks shown in Fig. 3.10. Application of a paired Student t -test per track revealed no statistically significant difference between the results of our algorithm and that of manual tracking, for both expert human observers ($p \gg 0.05$ in all cases). Often, biologists are interested in average velocities over sets of tracks. In the described experiments, the difference in average velocity (per 10 tracks) between automatic and manual tracking was less than 1%, for both observers. Our velocity estimates are also comparable to those reported previously based on manual tracking in the same type of image data [155].

Finally, we present two different example visualizations of real data together with the results of tracking using our algorithm. Fig. 3.11 shows the results of tracking in the presence of photobleaching, which clearly illustrates the capability of our algorithm to initiate new tracks for appearing objects, to terminate tracks for disappearing objects, and to deal with closely passing objects. The rendering in Fig. 3.12 gives a

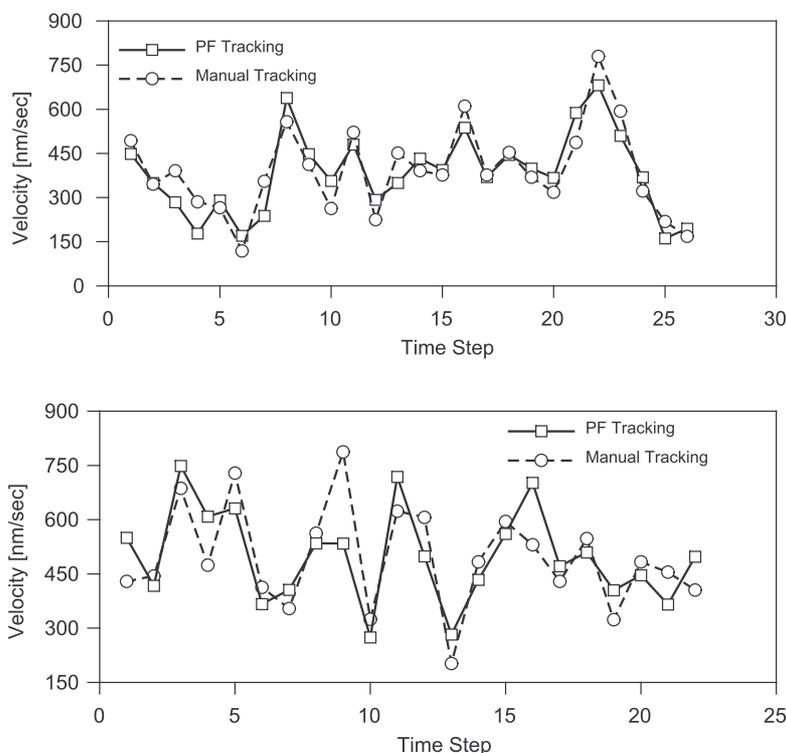


Figure 3.10. Velocity estimates per time step for our automatic tracking algorithm versus manual tracking. Results are shown for track numbers 4 (top) and 10 (bottom) in Fig. 3.9 (also from the top and bottom graphs, respectively).

visual impression of the full tracking results for a few time frames of one of the real data sets used in the experiments.

3.6 Discussion and Conclusions

In this chapter we have demonstrated the applicability of particle filtering for quantitative analysis of subcellular dynamics. Compared to existing approaches in this field, our approach is a substantial improvement for detection and tracking of large numbers of spots in image data with low SNR. Conventional methods, which perform object detection prior to the linking stage, use non-Bayesian maximum likelihood or least squares estimators. The variance of those estimators is larger than the variance of the MMSE estimator [11], for which some prior information about the estimated parameters is assumed to be known. In our case, this information is the prediction of the object position according to the motion model. This step, which optimally exploits available temporal information, makes our probabilistic tracking approach

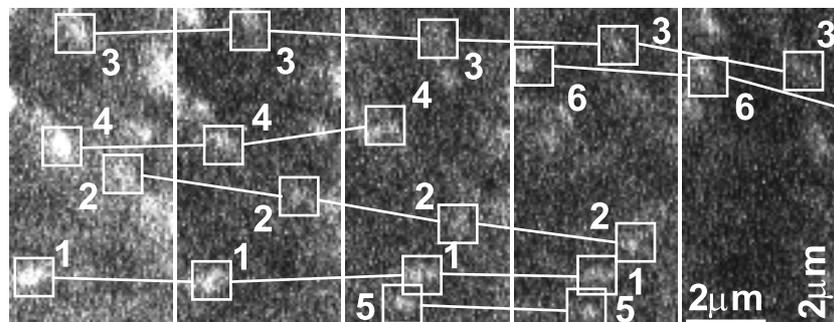


Figure 3.11. Results (six tracks) of automatically tracking MTs (bright spots) in the presence of photobleaching, illustrating the capability of our algorithm to capture newly appearing objects (tracks 5 and 6) and to detect object disappearance (for example track 4). It also shows the robustness of the algorithm in the case of closely passing objects (tracks 1 and 5).

perform superior in the presence of severe noise in comparison with existing frame-by-frame approaches, which break down at $\text{SNR} < 4\text{--}5$ [26, 32]. As the experiments show, contrary to two other popular tracking tools, our algorithm still yields reliable tracking results even in data with SNR as low as 2 (which is not uncommon in practice). We note that the comparison with these two-stage tracking approaches mainly evaluated the linking parts of the algorithms, as the detection part is based on thresholding, and the parameters for that stage were optimized manually until all the desired objects were localized. In practice, since these algorithms were not designed specifically to deal with photobleaching effects, they can be expected to perform worse than reported here.

The results of the experiments on synthetic image data suggest that our algorithm is potentially more accurate than manual tracking by expert human observers. The experiments on real fluorescence microscopy image sequences from MT dynamics studies showed comparable performance. This is explained by the fact that in the latter experiments, we were limited to comparing distributions and averages (Figs. 3.8 and 3.9), which may conceal small local discrepancies, especially when the objects' velocities vary over time. Instant velocities were also analyzed per track (Fig. 3.10) but could not be quantitatively validated due to the lack of ground truth. Nevertheless, the results indicate that our algorithm may replace laborious manual procedures. Currently we are evaluating the method also for other biological applications to further demonstrate its advantages over current means of manual and automated tracking and quantification of subcellular dynamics. Our findings encourage use of the method to analyze complex biological image sequences not only for obtaining statistical estimates of average velocity and life span, but also for detailed analyses of complete life histories.

The algorithm was implemented in the Java programming language (Sun Microsystems Inc., Santa Clara, CA) as a plugin for ImageJ (National Institutes of Health, Bethesda, MD [121]), a public domain and platform independent image pro-

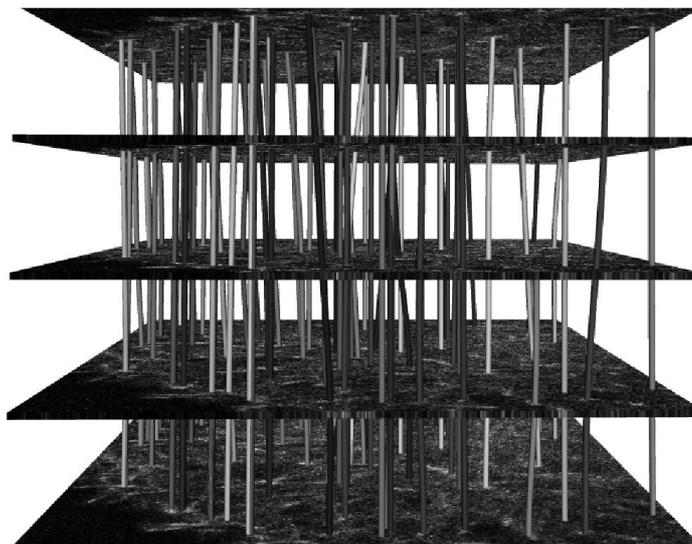
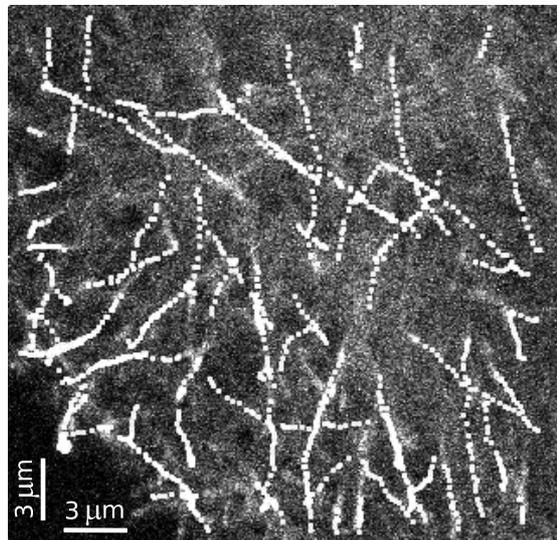


Figure 3.12. Visualization of tracking results (80 tracks) produced by our algorithm in the case of the real fluorescence microscopy image sequence of Fig. 3.1(a). Left: Trajectories projected on top of one of the frames, giving an impression of the MT dynamics in this image sequence. Right: Five frames from the sequence (time is increasing from bottom to top) with the trajectories rendered as small tubes connecting the frames. The rendering was accomplished using a script developed in-house based on the Visualization Toolkit [136].

cessing program used abundantly in biomedical image analysis [1]. Running on a regular PC (a Pentium IV with 3.2 GHz CPU and 3 GB of RAM) using the Java Virtual Machine version 1.5, the processing time per object per frame using 10^3 MC particles is about 0.3 sec. This cost is independent of image size, because all computations are done only for measurements falling inside the gates (defined for each track). We expect that faster execution times are still possible, after further optimization of the code. In the near future the algorithm will be integrated into a user-friendly software tool which will be made publically available.

The recursive nature of the proposed method (only the measurements up to time t are required in order to estimate the object positions at time t) can be effectively utilized to dramatically increase the throughput of live cell imaging experiments. Usually time-lapse imaging requires constant adjustment of the imaging field and focus position to keep the cell of interest centered in the imaged volume. There are basically two methods to track moving objects with a microscope. Most commonly, images are acquired at a fixed stage and focus position and the movements are analyzed afterwards, using batch image processing algorithms. The second possibility, rarely implemented, is to program the microscope to follow the movements of the cell automatically and keep it in the field of view. Such tracking systems have been developed previously [82, 119, 120], but they are either hardware-based or not easily portable to other microscopes. Using the proposed software-based tracking method, however, it can be implemented on any fluorescence microscope with motorized stage and focus. The prediction step of the algorithm can be used to adapt the field of view and steer the laser in the direction of moving objects. This also suggests a mechanism for limiting laser excitation and thereby reducing photobleaching.