Bioinformatics, 33(7), 2017, 1073–1080
doi: 10.1093/bioinformatics/btw751
Advance Access Publication Date: 4 January 2017
Original Paper

OXFORD

Bioimage informatics

Automated neuron tracing using probability hypothesis density filtering

Miroslav Radojević and Erik Meijering*

Biomedical Imaging Group Rotterdam, Departments of Medical Informatics and Radiology, Erasmus University Medical Center, 3000 CA Rotterdam, The Netherlands

*To whom correspondence should be addressed.

Associate Editor: Robert Murphy

Received on July 28, 2016; revised on October 19, 2016; editorial decision on November 20, 2016; accepted on November 22, 2016

Abstract

Motivation: The functionality of neurons and their role in neuronal networks is tightly connected to the cell morphology. A fundamental problem in many neurobiological studies aiming to unravel this connection is the digital reconstruction of neuronal cell morphology from microscopic image data. Many methods have been developed for this, but they are far from perfect, and better methods are needed.

Results: Here we present a new method for tracing neuron centerlines needed for full reconstruction. The method uses a fundamentally different approach than previous methods by considering neuron tracing as a Bayesian multi-object tracking problem. The problem is solved using probability hypothesis density filtering. Results of experiments on 2D and 3D fluorescence microscopy image datasets of real neurons indicate the proposed method performs comparably or even better than the state of the art.

Availability and Implementation: Software implementing the proposed neuron tracing method was written in the Java programming language as a plugin for the ImageJ platform. Source code is freely available for non-commercial use at https://bitbucket.org/miroslavradojevic/phd.

Contact: meijering@imagescience.org

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Accurate reconstruction of the tree-like structure of neuronal cells from optical microscopy images is a crucial step in automating the analysis of single neuron morphology or the connectivity of neuronal networks (Donohue and Ascoli, 2011; Meijering, 2010; Peng *et al.*, 2015). Microscopic images provide detailed information about the geometrical and topological properties of the neuronal arbors. Extracting and representing this information in a faithful and convenient digital format is key to many studies (Ascoli, 2002; Ascoli *et al.*, 2007; Halavi *et al.*, 2012; Lu *et al.*, 2015; Senft, 2011; Svoboda, 2011), as digital reconstructions enable neurobiologists to use computational approaches in addressing open issues in brain research, such as the relation between neuron structure and function, and the effects of neurodegenerative disease processes and drug compounds on neuron development and connectivity.

Existing approaches to tracing neurons in images can be broadly divided into global and local approaches. Global approaches consider the problem from the whole-image perspective and typically involve global image segmentation (Basu *et al.*, 2013; De *et al.*, 2016; Wearne *et al.*, 2005) or global optimization strategies (Türetken *et al.*, 2011; Xiao and Peng, 2013). Local approaches, on the other hand, use local image exploration strategies starting from seed points (Choromanska *et al.*, 2012; Peng *et al.*, 2011; Yang *et al.*, 2013) to find segments of the neuronal tree, which are then merged into a full tree representation. Both approaches have advantages and disadvantages and they are often combined to profit from their complementarity (Jiménez *et al.*, 2015; Zhao *et al.*, 2011).

A wide variety of computational concepts have been proposed in developing automated neuron tracing methods, whether global or local (Acciai *et al.*, 2016). These include active contours (Cai *et al.*, 2006;

© The Author 2017. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com

Luo *et al.*, 2015; Wang *et al.*, 2011), tubular models (Santamaría-Pang *et al.*, 2015), principal curves (Bas and Erdogmus, 2011; Quan *et al.*, 2016), perceptual grouping (Narayanaswamy *et al.*, 2011), path pruning (Peng *et al.*, 2011; Xiao and Peng, 2013), critical point detection (Al-Kofahi *et al.*, 2008; Radojević *et al.*, 2016), voxel scooping (Rodriguez *et al.*, 2009), dynamic and integer programming (Türetken *et al.*, 2012; Zhang *et al.*, 2007), active learning (Gala *et al.*, 2014), graph optimization (Chothani *et al.*, 2011; Türetken *et al.*, 2011), tubularity flow field segmentation (Mukherjee *et al.*, 2015), marked point processes (Basu *et al.*, 2016), iterative back-tracking (Liu *et al.*, 2016), and more. Space limitations do not permit a full discussion of all these concepts, but a key characteristic relevant to the present paper is that the vast majority of them are deterministic by nature. That is, they utilize models and algorithms that always assume or pass through the exact same sequence of states. While this behavior may seem virtuous and practically convenient, it is nonetheless not very realistic and not necessarily advantageous, for several reasons. For starters, expert human annotators, which are still considered to be the gold standard in evaluating methods, do not operate deterministically: their output will be (slightly) different every time they repeat a task. Also, any deterministic model is typically a (gross) simplification of reality, and consequently lacks flexibility in dealing with data variability. Finally, since every run of a deterministic algorithm will yield exactly the same output, it is not possible to accumulate evidence from multiple iterations.

In this paper we propose a new method for neuron tracing in optical microscopy images that operates probabilistically rather than deterministically. Focusing on delineating the branch centerlines, it utilizes a Bayesian approach to blend two sources of information: the model (based on prior knowledge) and the measurements (from the image data). The main novelty is that it combines the problems of neuron segment detection and linking into one framework by performing simultaneous multi-object tracking. Traditional multi-object (also referred to as multi-target) tracking techniques (Mahler, 2007; Stone *et al.*, 2013) typically assume the number of objects to be known and/or they explicitly associate measurements with objects which are then Bayesian filtered individually (Bar-Shalom and Li, 1995). Since in our application the number of objects (neuron segments) is unknown a priori, we use a different approach, based on filtering the so-called probability hypothesis density (PHD) function (Mahler, 2003). PHD filtering has gained popularity in recent years as a robust approach to tracking, since it is able to compensate for missing detections and to remove noise and clutter, while reducing the computational complexity from exponential to linear as the number of objects grows. Applications include radar and sonar tracking (Clark *et al.*, 2007; Tobias and Lanterman, 2005), video surveillance (Maggio *et al.*, 2008; Wang *et al.*, 2008) and even motion tracking in microscopy (Schlangen *et al.*, 2016; Wood *et al.*, 2012), but to the best of our knowledge it has not been explored yet for neuron tracing. Moreover, our application differs fundamentally from other works in the sense that the filtering is applied in space rather than in time. The proposed method is evaluated on a variety of real image data (both 2D and 3D) taking expert manual annotations as the gold standard. Its performance is compared with several state-of-the-art tools for neuron tracing (Chothani *et al.*, 2011; Quan *et al.*, 2016; Xiao and Peng, 2013).

2 Methods

2.1 Multi-object Bayesian filtering

We consider single-object tracking as a Bayesian inference problem (Bar-Shalom *et al.*, 2001; Särkkä, 2013). The key idea is to estimate the posterior probability density function (pdf) $f_{k|k}(\mathrm{x}_k|\mathrm{z}_{1:k})$, where x_k denotes the object state at iteration k, and $\mathrm{z}_{1:k}$ the sequence of observations from iterations 1 to k inclusive. Estimation is accomplished by sequentially applying prior knowledge to predict the state in the next iteration and updating this estimate with available observations. Similarly, multi-object tracking can be formulated as the problem of updating predictions of the multi-object state $\mathrm{X}_k = \{\mathrm{x}_{k,1}, \ldots, \mathrm{x}_{k,N_k}\}$ with multi-object observations $\mathrm{Z}_k = \{\mathrm{z}_{k,1}, \ldots, \mathrm{z}_{k,M_k}\}$, where N_k and M_k denote the number of objects and observations at iteration k, respectively. Formally:

$$\text{Prediction}: \quad f_{k|k-1}(\mathrm{X}_k|\mathrm{Z}_{1:k-1}) = \int \Pi_{k|k-1}(\mathrm{X}_k|\mathrm{X}_{k-1}) f_{k-1|k-1}(\mathrm{X}_{k-1}|\mathrm{Z}_{1:k-1}) \delta\mathrm{X}_{k-1} \tag{1}$$

$$\text{Update}: \quad f_{k|k}(\mathrm{X}_k|\mathrm{Z}_{1:k}) = \frac{\vartheta_k(\mathrm{Z}_k|\mathrm{X}_k) f_{k|k-1}(\mathrm{X}_k|\mathrm{Z}_{1:k-1})}{\int \vartheta_k(\mathrm{Z}_k|\mathrm{X}) f_{k|k-1}(\mathrm{X}|\mathrm{Z}_{1:k-1}) \delta\mathrm{X}} \tag{2}$$

where $\Pi_{k|k-1}(\mathrm{X}_k|\mathrm{X}_{k-1})$ denotes the multi-object state transition probability and $\vartheta_k(\mathrm{Z}_k|\mathrm{X}_k)$ the multi-object likelihood. Filtering the multi-object posterior $f_{k|k}(\mathrm{X}_k|\mathrm{Z}_{1:k})$ suffers from serious practical obstacles, as the multi-object state can be very high-dimensional and hard to sample and integrate efficiently. Moreover, it is necessary to take into account changes in object numbers, which adds an often intractable combinatorial burden. Thus more feasible solutions are needed.

2.2 Probability hypothesis density filtering

To overcome the difficulties of direct multi-object Bayesian filtering, we propose instead to filter the first-order statistical moment of the multi-object posterior $f_{k|k}(\mathrm{X}_k|\mathrm{Z}_{1:k})$, computed as

$$D_{k|k}(\mathrm{x}|\mathrm{Z}_{1:k}) = \int \delta_{\mathrm{X}}(\mathrm{x}) f_{k|k}(\mathrm{X}|\mathrm{Z}_{1:k}) \delta\mathrm{X} \tag{3}$$

where δ_{X} denotes the sum of Dirac deltas at elements of X. For the sake of notational convenience we abbreviate the left-hand side of (3) to $D_{k|k}(\mathrm{x})$ in the sequel. This function, known as the probability hypothesis density (PHD) (Mahler, 2003), is a non-negative function whose integral $\int D_{k|k}(\mathrm{x})\mathrm{dx}$ yields the expected number of objects $\nu_k \in \mathbb{R}$. PHD filtering allows for joint detection and estimation of an unknown and varying number of objects and their individual states using the Bayesian prediction and update framework. Here, multi-object state X_k and observation Z_k are modeled as so-called random finite sets (RFS), with randomness in set size as well as set element values (Bar-Shalom and Li, 1995), accommodating phenomena such as object initiation, clutter and partitioning (spawning). Formally stated, PHD filtering proceeds as follows:

$$\begin{aligned}\text{Prediction}: \quad & D_{k|k-1}(\mathrm{x}) = \gamma_{k|k-1}(\mathrm{x}) \\ & + \langle \beta_{k|k-1}(\mathrm{x}|\cdot) + p_{S,k|k-1}(\cdot)\pi_{k|k-1}(\mathrm{x}|\cdot), D_{k-1|k-1}(\cdot)\rangle\end{aligned} \tag{4}$$

$$\begin{aligned}\text{Update}: \quad D_{k|k}(\mathrm{x}) = {} & (1 - p_{D,k}(\mathrm{x}))D_{k|k-1}(\mathrm{x}) \\ & + \sum_{\mathrm{z}\in Z_k} \frac{p_{D,k}(\mathrm{x})g_k(\mathrm{z}|\mathrm{x})D_{k|k-1}(\mathrm{x})}{C_k(\mathrm{z}) + \langle p_{D,k}(\cdot)g_k(\mathrm{z}|\cdot), D_{k|k-1}(\cdot)\rangle}\end{aligned} \tag{5}$$

where $\gamma_{k|k-1}$ denotes the intensity function of newborn objects from iteration $k-1$ to k, $\beta_{k|k-1}$ the spawning object transition density, $p_{S,k|k-1}$ the object survival probability, $\pi_{k|k-1}$ the single-object transition density, $p_{D,k}$ the object detection probability, g_k the

single-object likelihood, C_k the clutter intensity function, and $\langle g(\cdot), f(\cdot)\rangle \equiv \int f(\xi)g(\xi)\mathrm{d}\xi$ (see e.g. Vo and Ma (2006) for details). An analytical solution to (4)-(5) is provided by the Gaussian-mixture PHD (GM-PHD) filter (Vo and Ma, 2006) but it is based on linear Gaussian assumptions regarding object birth and dynamics. A more general solution is offered by sequential Monte-Carlo PHD (SMC-PHD) filtering (Ristic *et al.*, 2010; Vo *et al.*, 2005; Zajic and Mahler, 2003), which approximates the PHD with a set of N random particles $\mathrm{x}^n_{k|k}$ and corresponding weights $\omega^n_{k|k}$ as

$$D_{k|k}(\mathrm{x}) \approx \sum_{n=1}^{N} \omega^n_{k|k}\delta_{\mathrm{x}^n_{k|k}}(\mathrm{x}) \tag{6}$$

so that the classic particle filtering scheme (Arulampalam *et al.*, 2002; Doucet *et al.*, 2000; Ristic *et al.*, 2004) can be applied.

2.3 PHD-filtering based neuron tracing

2.3.1 Definition and initialization

The multi-object filtering scheme we propose for neuron tracing defines the object state as an oriented location:

$$\mathrm{x} = [\mathrm{p}_\mathrm{x}, \mathrm{v}_\mathrm{x}] = [x, y, z, v_x, v_y, v_z] \tag{7}$$

where $\mathrm{p}_\mathrm{x} = [x, y, z]$ denotes the location and $\mathrm{v}_\mathrm{x} = [v_x, v_y, v_z]$ the local orientation of a tubular segment. Filtering starts from a set of N_0 seeds (Fig. 1) sampled from a seed pool consisting of the local maxima of the tubularity image $\tau(x, y, z)$ computed from the original image using Hessian-based multiscale line filtering (Sato *et al.*, 1998) and min–max normalized to $[0, 1]$. Local maxima are sorted in descending order so that seeds with high tubularity (meaning high confidence in the underlying image structure being a neuron branch) are processed first. To avoid seeds being selected too close together, in other words to ensure good spatial coverage of the neuron with seeds, for each selected seed (while going from top to bottom of the sorted list) the seeds within a circular neighborhood with radius r_0 are ignored in the current round. If, after SMC-PHD filtering (described next), the seed pool is not exhausted, a new round is started by selecting a new set of seeds. During filtering, the observation consists of the location and corresponding tubularity value:

$$\mathrm{z} = [\mathrm{p}_\mathrm{z}, \tau_\mathrm{z}] = [x, y, z, \tau] \tag{8}$$

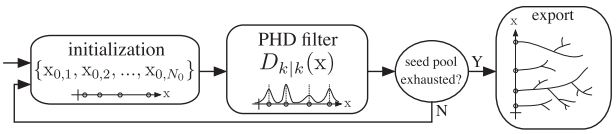

Fig. 1. Method overview. Each multi-object filtering round is initialized with N_0 seeds. If the seed pool is not exhausted by the end of the current round, a new round is started, and this is repeated until all seeds have been processed

2.3.2 SMC-PHD algorithm

The proposed method implements neuron tracing by SMC-PHD filtering. It is based on an approximation of $D_{k|k}(\mathrm{x})$ in (6) using $N = \rho N_k$ particles, where N_k denotes the number of objects to be filtered, and ρ the number of particles per object. That is, the state of object i at iteration k, denoted $\mathrm{x}_{k,i}$, is represented by ρ random particles $\mathrm{x}^n_{k|k}$ with corresponding weights $\omega^n_{k|k}$ (Fig. 2A). The multi-object state transition in the prediction step (4) is a collection of single-object transitions (Fig. 2B) that are approximated with transitions at the particle level (Fig. 2C). More specifically, at the initial iteration $k = 0$, N_0 seeds are selected and ρ particles are sampled in a circular neighborhood with radius r_0 around each seed location using the tubularity value for importance sampling to determine the weights, resulting in the weighted particle set $\{\omega^n_{0|0}, \mathrm{x}^n_{0|0}\}_{n=1}^{\rho N_0}$. The initial local orientation of each particle, $\mathrm{v}_{\mathrm{x}^n_{0|0}}$, is the unit vector pointing from the seed location to the particle location $\mathrm{p}_{\mathrm{x}^n_{0|0}}$. Subsequently, the prediction (4) and update (5) steps are executed for iterations $k = 1, 2, 3, \ldots$, until convergence. The transition and observation models (described next) allow to incorporate application-specific knowledge in this process. At iteration k, the set of weighted particles $\{\omega^n_{k-1|k-1}, \mathrm{x}^n_{k-1|k-1}\}_{n=1}^{\rho N_{k-1}}$ from iteration $k-1$ is used to predict η new particles for each persistent and spawned object (Fig. 2C). In the update step (5), a set of observations $\{\mathrm{z}_{k,j}\}_{j=1}^{M_k}$ is used to update the predicted particle weights, followed by estimation of the states $\{\widehat{\mathrm{x}}_{k,i}\}_{i=1}^{N_k}$. For details we refer to the algorithm pseudo codes in the supplementary information.

2.3.3 Transition model

In the prediction step (4), three types of objects are assumed: newborn, persisting and spawned objects (Vo *et al.*, 2005; Vo and Ma, 2006). In our algorithm, by design (since we use seeding), newborn objects are not considered, hence $\gamma_{k|k-1}(\mathrm{x}) = 0$.

Persisting objects in the current iteration correspond directly to existing objects in the previous iteration. In our algorithm, the

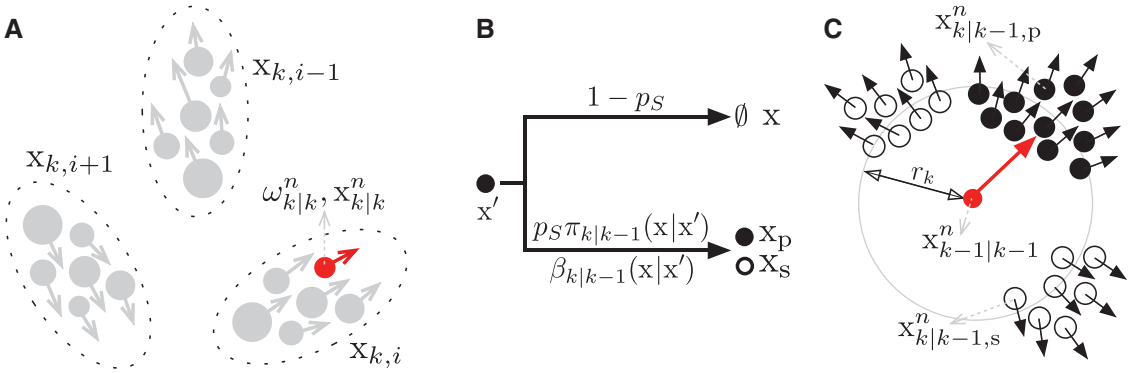

Fig. 2. PHD filtering using a particle representation. (**A**) Each object i at iteration k has a state $\mathrm{x}_{k,i}$ that is represented by random particles $\mathrm{x}^n_{k|k}$ with corresponding weights $\omega^n_{k|k}$. (**B**) In the transition from iteration $k-1$ to k an object (x') may disappear ($\emptyset$), persist (x_p), or spawn (x_s) according to the corresponding transition functions. Here p_S is shorthand notation for $p_{S,k|k-1}(\mathrm{x}')$, since in practice a constant is used (Table 1). (**C**) For each particle a prediction $\mathrm{x}^n_{k-1|k-1} \rightarrow \mathrm{x}^n_{k|k-1}$ is made within radius r_k according to the transition functions for persistence (p) and spawning (s)

transition density for predicting persistent object x given object x′ in the previous iteration, is calculated as

$$\pi_{k|k-1}(\mathrm{x}|\mathrm{x}';\kappa,r_k)=\frac{1}{\tilde{\pi}}e^{\frac{-(|\mathrm{p_x}-\mathrm{p_{x'}}|-r_k)^2}{2(r_k/3)^2}}\frac{e^{\kappa(\mathrm{v_x}\cdot\mathrm{v_{x'}})}}{2\pi I_0(\kappa)} \tag{9}$$

where $\tilde{\pi}$ is a normalization factor such that the sum of $\pi_{k|k-1}$ over $|\mathrm{p_x}-\mathrm{p_{x'}}|\leq 2r_k$ is unity, and I_0 is the zero-order Bessel function of the first kind. The first factor corresponds to a radial profile that peaks at the prediction step size r_k. The second factor is a circular normal distribution (von Mises) parametrized with the unit direction vector $\mathrm{v_{x'}}$ from the previous iteration and circular variance κ. Here, $\mathrm{v_x}=(\mathrm{p_x}-\mathrm{p_{x'}})/|\mathrm{p_x}-\mathrm{p_{x'}}|$, which connects the predicted location $\mathrm{p_x}$ with the location $\mathrm{p_{x'}}$ from the previous iteration. Particles $\mathrm{x}^n_{k|k-1,\mathrm{p}}$ (Fig. 2C) are drawn using $\pi_{k|k-1}$ as importance sampling function.

A spawned object is a new instance derived (spawned) from an existing object in the previous iteration. This allows dealing with bifurcations during tracing. In our algorithm, the transition density for predicting a spawned object x given x′ in the previous iteration, is calculated as

$$\beta_{k|k-1}(\mathrm{x}|\mathrm{x}';\kappa,r_k)=\frac{1}{\tilde{\beta}}e^{\frac{-(|\mathrm{p_x}-\mathrm{p_{x'}}|-r_k)^2}{2(r_k/3)^2}}\cdot\prod_{i=0}^{1}\left(1-\frac{e^{\kappa(-1^i\mathrm{v_x}\cdot\mathrm{v_{x'}})}}{2\pi I_0(\kappa)}\right) \tag{10}$$

where $\tilde{\beta}$ is a normalization factor such that the sum of $\beta_{k|k-1}$ over $|\mathrm{p_x}-\mathrm{p_{x'}}|\leq 2r_k$ is unity. The first factor has the same form as in (9) and the second factor is the aggregate of the complementary circular normal distributions used for spawning objects in positive and negative direction. An example of the intensity profile of $\pi_{k|k-1}$ and $\beta_{k|k-1}$ is shown in Supplementary Figure S1. Particles $\mathrm{x}^n_{k|k-1,\mathrm{s}}$ (Fig. 2C) are drawn using $\beta_{k|k-1}$ as importance sampling function.

2.3.4 Observation model

In the update step (5), a set of observations $\{\mathrm{z}_{k,j}\}_{j=1}^{M_k}$ is used to update the predictions from (4). Observations have a corrective role as they carry information about the neuron centerline locations and corresponding tubularity values (8). In our algorithm we use

$$h(\mathrm{p}|\mathrm{x}';\kappa,r_k)=\frac{1}{\tilde{h}}e^{\frac{-(|\mathrm{p}-\mathrm{p_{x'}}|-r_k)^2}{2(r_k/3)^2}}\frac{e^{\kappa(\mathrm{v_p}\cdot\mathrm{v_{x'}})}}{2\pi I_0(\kappa)}\tau(\mathrm{p}) \tag{11}$$

as the importance sampling function to obtain the observations, where $\tilde{h}$ is a normalization factor such that the sum of h over $|\mathrm{p}-\mathrm{p_{x'}}|\leq 2r_k$ is unity, and $\mathrm{v_p}=(\mathrm{p}-\mathrm{p_{x'}})/|\mathrm{p}-\mathrm{p_{x'}}|$. The first two factors have the same form as in (9) but here κ is typically lower to make the update step more restrictive than the prediction step. The third factor is the normalized tubularity measure τ (Sato *et al.*, 1998) at location p, which makes the observations correspond preferably to regions with high tubularity, which are indeed more likely to contain neuron structures.

To obtain the observations at iteration k, for each object i from the previous iteration a set of particles $\{\mathrm{p}^n_i\}_{n=1}^{\rho N_{k-1}}$ is drawn from h using $\mathrm{x}'=\widehat{\mathrm{x}}_{k-1,i}$ (the object state estimate), with particle weight proportional to the tubularity value at that location. All these particles together are subsequently clustered in an unsupervised manner using mean-shifting (Cheng, 1995), resulting in a set of clusters $\{\mathcal{C}_j\}_{j=1}^{M_k}$, with each cluster $\mathcal{C}_j$ having a subset $\{\mathrm{p}^n_{i,j}\}_{n=1}^{|\mathcal{C}_j|}$ of the particles. For each cluster, a representative sample $\mathrm{p}^{\hat{n}}_{i,j}$ is calculated using least-squares optimization,

$$\hat{n}=\arg\min_n\sum_{m\in[1,|\mathcal{C}_j|]}\theta(\mathrm{p}^m_{i,j},\mathrm{p}_{\widehat{\mathrm{x}}_{k-1,i}},\mathrm{p}^n_{i,j}) \tag{12}$$

where $\theta(\mathrm{p}_0,\mathrm{p}_1,\mathrm{p}_2)$ denotes the squared Euclidean distance from point p_0 to the line segment defined by p_1 and p_2, calculated as

$$\theta(\mathrm{p}_0,\mathrm{p}_1,\mathrm{p}_2)=\begin{cases}|\mathrm{p}_0-\mathrm{p}_1|^2 & \text{if } (\mathrm{p}_0-\mathrm{p}_1)\cdot(\mathrm{p}_2-\mathrm{p}_1)\leq 0\\ |\mathrm{p}_0-\mathrm{p}_2|^2 & \text{if } (\mathrm{p}_0-\mathrm{p}_2)\cdot(\mathrm{p}_1-\mathrm{p}_2)\leq 0\\ \dfrac{|(\mathrm{p}_2-\mathrm{p}_1)\times(\mathrm{p}_1-\mathrm{p}_0)|^2}{|\mathrm{p}_2-\mathrm{p}_1|^2} & \text{otherwise}\end{cases} \tag{13}$$

so that the line segment that best fits the cluster elements determines the selected location. From this the observation is obtained as $\mathrm{z}_{k,j}=[\mathrm{p}^{\hat{n}}_{i,j},\tau(\mathrm{p}^{\hat{n}}_{i,j})]$. The process is illustrated in Supplementary Figure S2.

For the single-object likelihood in (5) we use a Gaussian function centered at the spatial location of the observation, $g_k(\mathrm{z}|\mathrm{x})=\exp(-|\mathrm{p_z}-\mathrm{p_x}|^2/2\sigma_\mathrm{z}^2)$, giving more importance to predictions closer to z. The clutter intensity function is defined as an exponential dependency on the observation tubularity value, $C_k(\mathrm{z})=\exp(-K_c\tau_\mathrm{z})$, implying that the clutter increases as the tubularity value goes to zero. In practice, clutter plays a role in detecting terminal points, causing tracings with low particle weights (due to their proximity to regions with low tubularity values) to not be resampled and thus dropped after the update step.

2.3.5 Implementation details

Algorithms 1 and 2 in the supplementary information provide a step-by-step overview of our PHD-filtering based neuron tracing method. For testing purposes the method was implemented in Java as a plugin for ImageJ (Abràmoff *et al.*, 2004). The method has several parameters for which default parameters are given in Table 1. In our experience most of them do not require extensive tuning and for the experiments we used default values. An important aspect of any SMC-based algorithm is to use a sufficient number of particles in the approximations. In our experiments we found that values of 10–20 are sufficient for ρ and η since the objects of interest in our application (neurons) are approximately 1D structures in 3D space and therefore are easily covered. Higher values can lead to higher accuracy and precision but at proportionally higher computational cost. The most important parameters are the numbers of seeds N_0 and rounds (Fig. 1) and in the experiments (described next) we have

Table 1. Parameters of the proposed method with their default values

Parameter	Default	Description
K_c	30	Clutter intensity function decay
N_0	20	Number of seed points per round
p_D	0.9	Object detection probability
p_S	0.9	Object survival probability
r_k	3 voxels	Radial estimation step size
ρ	≥10	Number of particles per object
η	≥10	Number of predictions per particle
κ	2	Circular variance in (9) & (10)
	0.5	Circular variance in (11)

In our implementation we used constants for the object detection probability $p_D=p_{D,k}$ and the object survival probability $p_S=p_{S,k|k-1}$.

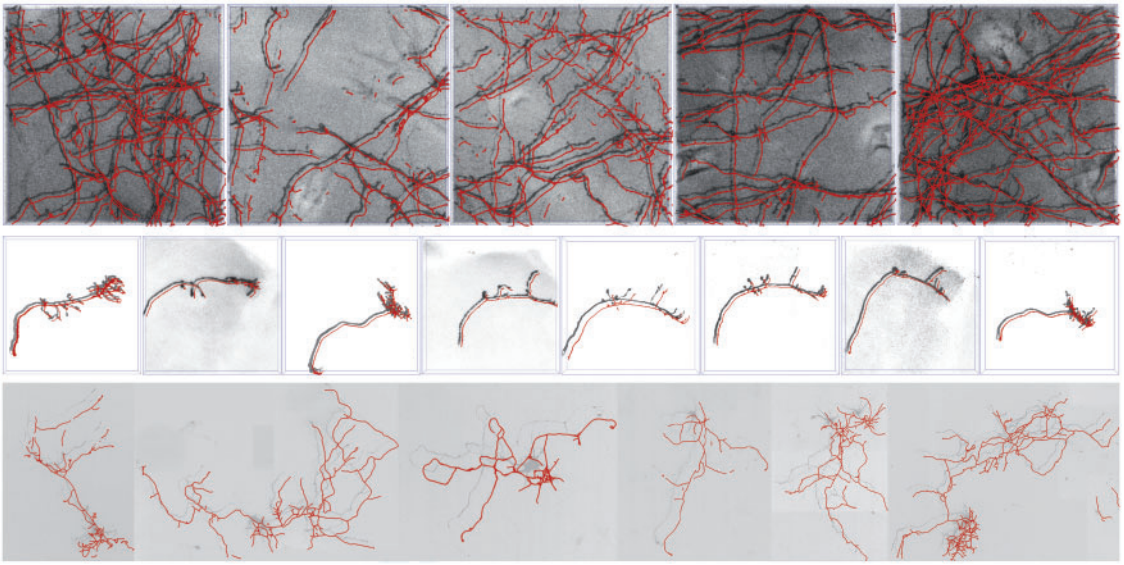

Fig. 3. Example images with tracing results of the datasets used in the evaluation. Top row: NCL1A image stacks (volume rendered) showing a network of neocortical layer-1 axons. Middle row: OPF image stacks (volume rendered) showing olfactory projection fibers. Bottom row: HCN images showing hippocampal neurons. The tracings (overlaid) were obtained with our method using 20 seeds and at most 10 rounds (up to 40 for the top row to capture more detail). For illustration purposes the image intensities are inverted in these visualizations compared to the originals, and the tracings are offset with respect to the neuron structures for better visual comparison (Color version of this figure is available at *Bioinformatics* online.)

tested the performance of our algorithm for different values of these parameters.

3 Results

3.1 Neuron datasets

For evaluating the performance of the proposed method for both 2D and 3D neuron tracing we used three datasets (Fig. 3) of real neuron images acquired with fluorescence microscopy. Two datasets are 3D image stacks from the DIADEM challenge (Brown *et al.*, 2011): neocortical layer-1 axons (NCL1A) with 16 image stacks and olfactory projection fibers (OPF) with 9 image stacks. The third dataset (HCN) consists of 30 2D images of hippocampal neurons (Steiner *et al.*, 2002). Together the datasets show a good variety of image contrast and structural complexity. We refer to the cited papers for further details about the images.

3.2 Performance measures

The accuracy of the tracings produced by our method was assessed by comparison with the gold-standard obtained by manual delineation of the neuron structures (Gillette *et al.*, 2011; Meijering *et al.*, 2004). To this end we used two categories of evaluation measures. The first consists of measures summarizing the spatial Euclidean distances between the nodes of two tracings to be compared: the average spatial distance (SD), the average substantial spatial distance (SSD), and the fraction of nodes whose distance is at least the substantial distance (%SSD). Similar to previous studies using these measures (Peng *et al.*, 2010) we set the substantial distance to 2 (pixels in 2D and voxels in 3D). The second category of evaluation measures are based on the numbers of true-positive (TP), false-positive (FP), and false-negative (FN) nodes according to the substantial distance. From these we compute the recall, $R = TP/(TP + FN)$, and precision, $P = TP/(TP + FP)$, summarized using the F-score, $F = 2PR/(P + R)$. Prior to computing these measures the tracings (from the method and the gold-standard) were resampled with an equal step size of 1 pixel using Vaa3D (Peng *et al.*, 2010).

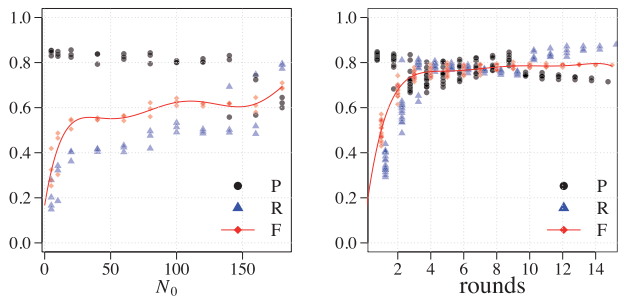

Fig. 4. Performance of our method as a function of numbers of seeds and rounds for an example image stack from the NCL1A dataset. Similar trends were observed for all stacks in the dataset. Left panel: Precision (P), recall (R) and F-score (F) after one round initialized with different numbers of seeds (N_0). Right panel: The scores after multiple rounds with a fixed number of seeds ($N_0 = 40$). Fifth-order polynomial fitting was used to show the approximate F-score trend (Color version of this figure is available at *Bioinformatics* online.)

3.3 Evaluation of method behavior

First we evaluated the behavior of our method as a function of the number of seeds and rounds. For this experiment we measured P, R and F for (i) a single round of filtering with different numbers of seeds and (ii) multiple rounds of filtering using a fixed number of seeds. Since our algorithm operates probabilistically we averaged the results of five repetitions of the experiment. The results for the NCL1A dataset are shown in Figure 4 and for the other datasets in Supplementary Figure S3. As expected, R and F generally increase, but P slightly decreases as the number of seeds and rounds increase, indicating an increase in the number of FP detections. The specific patterns may differ depending on the image content, but we observe that as a function of the number of seeds, the increase of R and F levels off beyond about 40, so we subsequently used this value. As a

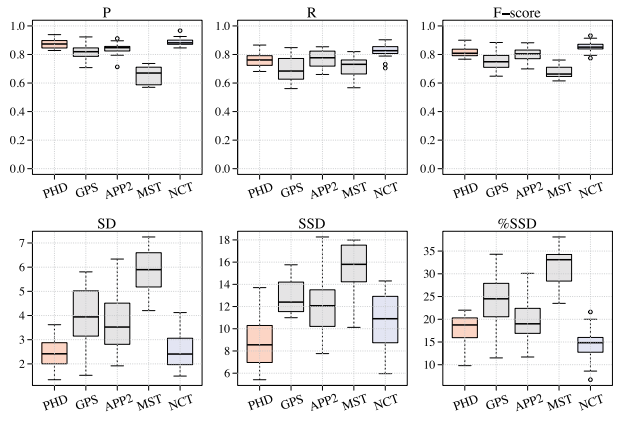

Fig. 5. Performance comparison of our method with several other methods on the NCL1A dataset. For each method and each measure, the plotted box indicates the 25-75 percentile, the horizontal bar indicates the median score, and the whiskers and outliers are drawn using the default settings of R (Color version of this figure is available at *Bioinformatics* online.)

function of the number of rounds, R and F level off after about 4 rounds, indicating there is no need in practice to run the method exhaustively on all possible seed points. This can be explained from the fact that seed selection proceeds from highest to lowest tubularity value, so that later seeds correspond to less and less valuable image structures, and the resulting tracings will be dropped due to low particle weights. Examples of traced neurons for the different datasets are shown in Figure 3. As can be observed from the examples in the top row of the figure, images with more fuzzy and fragmented structures may require more rounds to capture more detail. Alternatively, a better tubularity filter may be needed.

3.4 Comparison with other methods

Next we compared the performance of our method (PHD) with several alterative methods, namely all-path pruning (APP2) (Xiao and Peng, 2013), NeuroGPS-Tree (GPS) (Quan *et al.*, 2016), minimum spanning tree (MST) tracing as used in the BigNeuron project (Peng *et al.*, 2015), and Neural Circuit Tracer (NCT) (Chothani *et al.*, 2011). For each of these methods the scores were optimized by trying all possible parameter values on a grid. The results for the NCL1A dataset are shown in Figure 5 and for the other datasets in Supplementary Figures S4 and S5. We observe that our method (results indicated in red) performs comparably or even better than the state-of-the-art methods. This suggests there may indeed be an advantage in using probabilistic approaches such as the one proposed in this paper. We note that NCT (results indicated in Figure 5), while performing superiorly in most cases, required significant user interaction and manual correction to enable export of the tracings to the standard SWC file format used in our evaluations. Thus the results of this method include a high level of expert input and could serve as a reference. All other methods including our own were fully automatic after parameter selection.

To further demonstrate the advantage of our method over the others in challenging situations we also studied the case when neuron fibers meet, run closely parallel to each other for some distance, and then diverge again. In order to analyze the behavior of the different methods in a controlled manner, with increasing distance between the fibers, we synthesized images with two fibers of similar intensity and scale. The results, shown in Supplementary Figure S6, demonstrate that our PHD method, similar to GPS, yields more faithful tracings than APP2 and MST. NCT was not included in these experiments for reasons mentioned above. Not surprisingly, all methods break down when the fibers overlap completely. We also created an even more challenging case, with three fibers of different intensity and scale. The results, shown in Supplementary Figure S7, illustrate that our method outperforms even the best alternatives.

In terms of computational efficiency it turned out difficult to directly compare the methods. This was mainly due to the use of different programming languages (Java versus C++) and the varying efficiencies of underlying software libraries used on the different operating systems we considered (Linux Ubuntu and Mac OS). Moreover we observed that the absolute as well as the relative execution times of the different methods varied widely depending on the image content. Generally we found APP2 to be the fastest method (on the order of seconds per image), and PHD up to about one order of magnitude slower, while GPS and MST were either slower or faster than PHD depending on the configuration. NCT is ignored here for mentioned reasons. From these observations we conclude that the efficiency of our method is comparable to the state of the art.

4 Conclusions

We have presented a new method for tracing the branch centerlines of neurons based on Bayesian multi-object tracking using probability hypothesis density (PHD) filtering. The method is able to simultaneously trace out multiple neuron structures in a probabilistic fashion so that the same neuron segments may be covered multiple times and are thus supported by more evidence. PHD filtering solves the computational problems of direct Bayesian multi-object tracking and allows convenient handling of bifurcations and terminations during the tracing process by modeling of spawned objects and observation clutter. The results of experiments on various fluorescence microscopy image datasets of real neurons showed that the proposed method performs comparably or better than alternative state-of-the-art neuron tracing methods.

The current version of the proposed method is initialized with seed points sampled from the local maxima (from highest to lowest) of the tubularity filter response. This is a rather rudimentary approach that may sometimes result in missed branches (false negatives). Ideally, seeds should be strategically distributed so that they cover as many branches of the neuron structure as possible while avoiding background artifacts, and this is an important topic for further research. In addition, the current mechanism responsible for trace termination, based on the clutter term of the PHD filter, relies strongly on the tubularity score and thus is sensitive to local interruptions in neuron staining. This could be remedied by using a better tubularity filter and/or refining the clutter model. Thus, in future work, we will study whether further improvements could be achieved using different transition and observations models. We also aim to extend the method to perform local branch radius estimation during tracing in order to obtain complete neuron reconstructions.

Funding

This work was supported by the Netherlands Organization for Scientific Reseach (NWO) [grant number 612.001.018 awarded to EM].

Conflict of Interest: none declared.

References

Abràmoff,M.D. *et al*. (2004) Image processing with ImageJ. *Biophotonics Int*., **11**, 36–43.

Acciai,L. *et al*. (2016) Automated neuron tracing methods: an updated account. *Neuroinformatics*, **14**, 353–367.

Al-Kofahi,Y. *et al*. (2008) Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images. *Cytometry Part A*, **73**, 36–43.

Arulampalam,M.S. *et al*. (2002) A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process*., **50**, 174–188.

Ascoli,G.A. (2002) *Computational Neuroanatomy: Principles and Methods*. Humana Press, Totowa, NJ.

Ascoli,G.A. *et al*. (2007) NeuroMorpho.Org: a central resource for neuronal morphologies. *J. Neurosci*., **27**, 9247–9251.

Bar-Shalom,Y. and Li,X.R. (1995) *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT.

Bar-Shalom,Y. *et al*. (2001) *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. Wiley, New York.

Bas,E. and Erdogmus,D. (2011) Principal curves as skeletons of tubular objects: locally characterizing the structures of axons. *Neuroinformatics*, **9**, 181–191.

Basu,S. *et al*. (2013) Segmentation and tracing of single neurons from 3D confocal microscope images. *IEEE J. Biomed. Health Inf*., **17**, 319–335.

Basu,S. *et al*. (2016) Neurite tracing with object process. *IEEE Trans. Med. Imaging*, **35**, 1443–1451.

Brown,K.M. *et al*. (2011) The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, **9**, 143–157.

Cai,H. *et al*. (2006) Repulsive force based snake model to segment and track neuronal axons in 3D microscopy image stacks. *NeuroImage*, **32**, 1608–1620.

Cheng,Y. (1995) Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell*., **17**, 790–799.

Choromanska,A. *et al*. (2012) Automatic reconstruction of neural morphologies with multi-scale tracking. *Front. Neural Circuits*, **6**, 1–13.

Chothani,P. *et al*. (2011) Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, **9**, 263–278.

Clark,D. *et al*. (2007) Particle PHD filter multiple target tracking in sonar image. *IEEE Trans. Aerospace Electronic Syst*., **43**, 409–416.

De,J. *et al*. (2016) A graph-theoretical approach for tracing filamentary structures in neuronal and retinal images. *IEEE Trans. Med. Imaging*, **35**, 257–272.

Donohue,D.E. and Ascoli,G.A. (2011) Automated reconstruction of neuronal morphology: an overview. *Brain Res. Rev*., **67**, 94–102.

Doucet,A. *et al*. (2000) On sequential Monte Carlo sampling methods for Bayesian filtering. *Stat. Comput*., **10**, 197–208.

Gala,R. *et al*. (2014) Active learning of neuron morphology for accurate automated tracing of neurites. *Front. Neuroanat*., **8**,

Gillette,T.A. *et al*. (2011) DIADEMchallenge.Org: a compendium of resources fostering the continuous development of automated neuronal reconstruction. *Neuroinformatics*, **9**, 303–304.

Halavi,M. *et al*. (2012) Digital reconstructions of neuronal morphology: three decades of research trends. *Front. Neurosci*., **6**, 1–11.

Jiménez,D. *et al*. (2015) Improved automatic centerline tracing for dendritic and axonal structures. *Neuroinformatics*, **13**, 227–244.

Liu,S. *et al*. (2016) Rivulet: 3D neuron morphology tracing with iterative back-tracking. *Neuroinformatics*, **14**, 387–401.

Lu,Y. *et al*. (2015) Quantitative arbor analytics: unsupervised harmonic co-clustering of populations of brain cell arbors based on L-measure. *Neuroinformatics*, **13**, 47–63.

Luo,G. *et al*. (2015) Neuron anatomy structure reconstruction based on a sliding filter. *BMC Bioinformatics*, **16**, 342.

Maggio,E. *et al*. (2008) Efficient multitarget visual tracking using random finite sets. *IEEE Trans. Circuits Syst. Video Technol*., **18**, 1016–1027.

Mahler,R.P.S. (2003) Multitarget Bayes filtering via first-order multitarget moments. *IEEE Trans. Aerospace Electronic Syst*., **39**, 1152–1178.

Mahler,R.P.S. (2007) *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., Boston, MA.

Meijering,E. (2010) Neuron tracing in perspective. *Cytometry Part A*, **77**, 693–704.

Meijering,E. *et al*. (2004) Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A*, **58**, 167–176.

Mukherjee,S. *et al*. (2015) Tubularity flow field—a technique for automatic neuron segmentation. *IEEE Trans. Image Process*., **24**, 374–389.

Narayanaswamy,A. *et al*. (2011) 3-D image pre-processing algorithms for improved automated tracing of neuronal arbors. *Neuroinformatics*, **9**, 219–231.

Peng,H. *et al*. (2010) V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nat. Biotechnol*., **28**, 348–353.

Peng,H. *et al*. (2011) Automatic 3D neuron tracing using all-path pruning. *Bioinformatics*, **27**, i239–i247.

Peng,H. *et al*. (2015) BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images. *Neuron*. **87**, 252–256.

Quan,T. *et al*. (2016) NeuroGPS-Tree: automatic reconstruction of large-scale neuronal populations with dense neurites. *Nat. Methods*, **13**, 51–54.

Radojević,M. *et al*. (2016) Fuzzy-logic based detection and characterization of junctions and terminations in fluorescence microscopy images of neurons. *Neuroinformatics*, **14**, 201–219.

Ristic,B. *et al*. (2004) *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, Boston, MA.

Ristic,B. *et al*. (2010 Improved SMC implementation of the PHD filter. In: *Proceedings of the 13th Conference on Information Fusion*, pp. 1–8.

Rodriguez,A. *et al*. (2009) Three-dimensional neuron tracing by voxel scooping. *J. Neurosci. Methods*, **184**, 169–175.

Santamaría-Pang,A. *et al*. (2015) Automatic morphological reconstruction of neurons from multiphoton and confocal microscopy images using 3D tubular models. *Neuroinformatics*, **13**, 297–320.

Särkkä,S. (2013) *Bayesian Filtering and Smoothing*. Cambridge University Press, Cambridge, UK.

Sato,Y. *et al*. (1998) Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Med. Image Anal*., **2**, 143–168.

Schlangen,I. *et al*. (2016) Marker-less stage drift correction in super-resolution microscopy using the single-cluster PHD filter. *IEEE J. Select. Top. Signal Process*., **10**, 193–202.

Senft,S.L. (2011) A brief history of neuronal reconstruction. *Neuroinformatics*, **9**, 119–128.

Steiner,P. *et al*. (2002) Overexpression of neuronal Sec1 enhances axonal branching in hippocampal neurons. *Neuroscience*, **113**, 893–905.

Stone,L.D. *et al*. (2013) *Bayesian Multiple Target Tracking*. Artech House, Boston, MA.

Svoboda,K. (2011) The past, present, and future of single neuron reconstruction. *Neuroinformatics*, **9**, 97–98.

Tobias,M. and Lanterman,A.D. (2005) Probability hypothesis density-based multitarget tracking with Bistatic range and Doppler observations. *IEE Proc. Radar Sonar Navigation*, **152**, 195–205.

Türetken,E. *et al*. (2011) Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, **9**, 279–302.

Türetken,E. *et al*. (2012) Automated reconstruction of tree structures using path classifiers and mixed integer programming. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 566–573.

Vo,B.N. and Ma,W.K. (2006) The Gaussian mixture probability hypothesis density filter. *IEEE Trans. Signal Process*., **54**, 4091–4104.

Vo,B.N. *et al*. (2005) Sequential Monte Carlo methods for multitarget filtering with random finite sets. *IEEE Trans. Aerospace Electronic Syst*., **41**, 1224–1245.

Wang,Y. *et al*. (2011) A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, **9**, 193–217.

Wang,Y.D. *et al*. (2008) Data-driven probability hypothesis density filter for visual tracking. *IEEE Trans. Circuits Syst. Video Technol*., **18**, 1085–1095.

Wearne,S.L. *et al*. (2005) New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience*, **136**, 661–680.

Wood,T.M. *et al*. (2012) Simplified multitarget tracking using the PHD filter for microscopic video data. *IEEE Trans. Circuits Syst. Video Technol.*, **22**, 702–713.

Xiao,H. and Peng,H. (2013) APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, **29**, 1448–1454.

Yang,J. *et al*. (2013) A distance-field based automatic neuron tracing method. *BMC Bioinformatics*, **14**, 93.

Zajic,T. and Mahler,R.P.S. (2003) Particle-systems implementation of the PHD multitarget-tracking filter. In: *Proceedings of SPIE 5096: Signal Processing, Sensor Fusion, and Target Recognition XII*, pp. 291–299.

Zhang,Y. *et al*. (2007) Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays. *NeuroImage*, **35**, 1502–1515.

Zhao,T. *et al*. (2011) Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, **9**, 247–261.