APPLICATION INDEPENDENT GREEDY PARTICLE TRACKING METHOD FOR 3D FLUORESCENCE MICROSCOPY IMAGE SERIES

L. Paavolainen[1], P. Kankaanpää[2], P. Ruusuvuori[3,1], G. McNerney[4], M. Karjalainen[1] and V. Marjomäki[1]

[1]Department of Biological and Environmental Science, University of Jyväskylä, Jyväskylä, Finland
[2]Department of Biochemistry and Food Chemistry, University of Turku, Turku, Finland
[3]Department of Signal Processing, Tampere University of Technology, Tampere, Finland
[4]Center for Biophotonics Science and Technology, University of California Davis Medical Center, Sacramento, CA, USA

ABSTRACT

Single-particle tracking is computationally a challenging problem, and usually solved with local methods. Local methods suffer from defects in the image data or in the detection of particles, such as temporal disappearing of particles. A particle tracking method has to provide a solution also to real disappearing and appearing of particles as a result of merging and splitting. Here, we present an efficient, greedy algorithm as a solution to the particle tracking problem. This improved local method is application independent, as it has high configurability of the function used to solve particle correspondence. To demonstrate the accuracy of the method, we apply it to real microscopy image data with the BioImageXD software, validate it using simulated image data, and compare it to a well-known existing method.

Index Terms— Particle tracking, Biomedical image processing, Image motion analysis, Optical microscopy, Greedy algorithms

1. INTRODUCTION

In order to gain in-depth understanding of the cellular physiological processes during cell division, migration and pathological processes such as cancer and inflammation, it is imperative to study these processes in live cells. Recent advances in biomedical imaging have facilitated the detection of fast cellular processes in 3D.

The dynamic behaviour of particles ranging from molecular structures to whole cells, can be analysed using single-particle tracking methods. In single-particle tracking, the three-dimensional spatial locations of particles are traced through a time-lapse sequence of image volumes, producing particle trajectories that can be analysed for instance for speed or direction of movement.

However, single-particle tracking in fluorescence microscopy is a challenging task. First, the size, shape and intensity of fluorescent particles often vary through the recorded time series. This complicates the process of linking the detected particles, which there may be hundreds of, to the corresponding ones in the following time point. Second, estimating the particle location in the next time point may be problematic. Microscopic particles in cells often exhibit Brownian motion, causing their movements to be somewhat erratic, even if they are being actively transported. Larger biological particles, such as cells, also tend to constantly change their direction of movement and have a complex, labile morphology. For these reasons, many particle tracking methods search for the corresponding particle using a specified maximum distance from the location of the particle in the previous time point, instead of trying to estimate the direction of movement. Third, there are many events that can occur during the image series that complicate tracking [1]. In a *merge* event, two or more particles merge into one. This can happen when particles are closer to each other than what the resolution limit of the imaging device is, or when a real cluster is formed. A *split* event occurs when a single particle is split into two or more distinct particles. Additional events to take into account include *appearing* and *disappearing* of particles in the middle of a time series. Appearing of a completely new particle occurs when a particle has moved into focus or into the field of view, or when a particle has gained enough size and intensity as a result of the clustering of fluorophores. *Disappearing* is defined as an event where an individual particle disappears without merging with other particles.

General single-particle tracking problem is NP-hard [2], which makes finding a global solution, without approximations, computationally unfeasible even for tens of particles [1, 3]. To circumvent this constraint, many tracking algorithms solve particle correspondence locally. Greedy optimization makes particle tracking efficient, but it usually cannot give an optimal global solution to the particle tracking problem [4]. Also, some local algorithms do not handle merge and split events [5]. To improve local results, some algorithms use an additional global step to recover the temporal disappearing of particles [2, 3]. The downside is that many of these methods treat merge and split events as temporal disappearing and appearing of particles [2, 3].

In this paper, we present a new, greedy, local single-particle tracking method for 3D fluorescence microscopy image data, designed to overcome the challenges mentioned. Our algorithm handles merge, appear and disappear events using only one-pass through the particles in the time series. A split event is considered as the appearing of one new particle. We focus on finding trajectories of particles rather than methods to detect them. There are several methods for detecting particles, for instance spot detection methods for finding small point-like structures [6], or active contour methods for detecting large cells [7, 8]. The single-particle tracking method presented here is not application specific, and can thus be used to track the movement of virtually any type of particles.

2. METHODS

2.1. Particle tracking method

Proposed particle tracking method makes decisions using information of the existing trajectories and of the particles of the time point under evaluation. Optimization of the particle selection for each tra-

jectory is based on the calculation of a goodness value. The goodness value is calculated using particle location, size, average intensity and direction of movement, and information of previous particles in the trajectory. The user can control the method by setting weights to these four parameters. Additionally, the user can define terms of maximum deviation for the parameters that each particle has to fulfill before being considered for the trajectory. If required, the weights and terms can also be used to limit the tracking to specific trajectories, such as those of fairly straight moving particles.

The terms for size, average intensity and direction angle are defined using the specified maximum deviations, as compared to the previous particle in the trajectory. For the location term, the user can define a minimum (D_l) and a maximum (D_h) distance in addition to the maximum deviation. Let σ_d, σ_s, σ_i, $\sigma_a \geq 0$ be the user-defined deviations of distance, size, average intensity and direction angle, respectively. Also, let p_t be the time point, p_l the location, p_s the size, and p_i the average intensity of particle p, and let p_a be the angle of particle direction, and t_k the particle of k:th time point in trajectory t. Using these definitions, all particles p that fulfill the terms presented in equation 1 are considered to be included in the trajectory t.

$$\begin{aligned} D_l * (1 - \sigma_d) \leq d \leq D_h * (1 + \sigma_d) \\ \frac{|p_s - (t_{p_t-1})_s|}{(t_{p_t-1})_s} \leq \sigma_s \\ \frac{|p_i - (t_{p_t-1})_i|}{(t_{p_t-1})_i} \leq \sigma_i \\ \frac{|p_a - (t_{p_t-1})_a|}{(t_{p_t-1})_a} \leq \sigma_a \end{aligned} \quad (1)$$

where $d = \|p_l - (t_{p_t-1})_l\|$. The direction angle of particle p_a is calculated using the location of the particle p_l and the locations of two previous particles in the trajectory $(t_{p_t-1})_l$ and $(t_{p_t-2})_l$. Let $\vec{u}$ and $\vec{v}$ be vectors defined as $\vec{u} = p_l - (t_{p_t-1})_l$ and $\vec{v} = (t_{p_t-1})_l - (t_{p_t-2})_l$. The direction angle of the particle is then

$$p_a = \frac{\vec{u}}{||\vec{u}||} \cdot \frac{\vec{v}}{||\vec{v}||}.$$

To determine the goodness of fit for particle p in trajectory t, we define function Ψ separately for distance, size, average intensity and direction angle as defined in equation 2.

$$\begin{aligned} \Psi_d(t,p) &= \begin{cases} 1 & \text{if } D_l \leq d \leq D_h \\ \left| \frac{D_l - d}{D_l * \sigma_d} - 1 \right| & \text{if } d < D_l \\ \left| \frac{d - D_h}{D_h * \sigma_d} - 1 \right| & \text{if } d > D_h \end{cases} \\ \Psi_s(t,p) &= \left| \frac{|p_s - (t_{p_t-1})_s|}{(t_{p_t-1})_s * \sigma_s} - 1 \right| \\ \Psi_i(t,p) &= \left| \frac{|p_i - (t_{p_t-1})_i|}{(t_{p_t-1})_i * \sigma_i} - 1 \right| \\ \Psi_a(t,p) &= \left| \frac{|p_a - (t_{p_t-1})_a|}{(t_{p_t-1})_a * \sigma_a} - 1 \right| \end{aligned} \quad (2)$$

Finally, goodness of fit for particle p in trajectory t is computed using equation 3.

$$\Omega(t,p) = \sum_k w_k * \Psi_k(t,p) \quad (3)$$

Step 1. If first time point, create new trajectory for every particle in the time point and finish iteration.

Step 2. Calculate goodness value using equation 3 for every particle and track combination that fulfills definitions of equation 1.

Step 3. Go through each goodness value from highest to lowest. Add particle p to trajectory t, if t and p used to calculate goodness value are not used yet, t is not terminated, and $\Omega(t,p) > \alpha * h_t$, where h_t is the highest goodness value calculated for t. Mark every trajectory, that had same particle as trajectory t in previous time point, as terminated.

Step 4. Add particle with highest goodness of fit for every track that is not used or terminated. If there are no particles that fulfill definitions of equation 1, then mark trajectory as terminated.

Step 5. For every particle that is not used yet, create new track and add particle to that track.

Fig. 1. Description of the steps of our particle tracking algorithm, iterated for every time point. $\alpha \in [0,1]$ is a user-defined clustering factor. Higher α value results in more merging of trajectories.

where w_k, $k \in \{d, s, i, a\}$ are the user-defined weights for the four parameters. For the weights it holds that $w_k \in [0,1]$ and $\sum_k w_k = 1$.

The proposed particle tracking method creates trajectories using a breadth-first search of particles. Unlike depth-first search, this makes it possible to optimize the selection of the particles for all trajectories at every time point. The key steps of the particle tracking method, iterated for every time point, are presented in Figure 1. Step 1 is initialization, done only for the first time point. Step 2 calculates the goodness of fit of the particles to existing trajectories. In step 3, unused particles are assigned to unused trajectories. A trajectory does not necessarily get the particle that has the highest goodness value calculated in step 2, since that particle might already be assigned to another trajectory for which it had a higher goodness of fit. Clustering of the particles is handled in step 4, where also trajectories without suitable particles are terminated. This provides a solution to the *merge* and *disappear* events. Finally, in step 5, new trajectories are created for all unused particles. This step solves the *appear* and *split* events. An open source implementation of the proposed particle tracking method is available in the BioImageXD software [9].

2.2. Real data

We tested our particle tracking method with the known phenomenon of echovirus 1 induced clustering of its receptor, $\alpha 2\beta 1$ integrin [10]. The integrins were labeled with a fluorophore-conjugated antibody, and five consecutive image series were acquired, each consisting of 40 three-dimensional image stacks taken with 30 second intervals with a confocal microscope (Olympus FV1000; slice dimensions 512 x 512, number of slices per stack 19–22, voxel size 0.207 x 0.207 x 0.500 μm^3).

Before tracking, particles were detected from all 200 image stacks with an automatic spot detection method as follows: 1) difference of Gaussian filtering with 3D kernel, 2) voxel classification in three groups with Otsu's automatic thresholding [11], 3) voxels in

the brightest group labeled as particles with connected component labeling (removing particles smaller than seven voxels), and 4) particles quantified for centroid, volume and average intensity (values used in subsequent tracking). The number of detected particles per image stack ranged from 90 to 265. All processing was done with the BioImageXD software [9].

Trajectories were created using the particle tracking method with following parameters: $D_l = 0.0\mu$m, $D_h = 1.0\mu$m, $\sigma_d = 1.0$, $\sigma_s = 1.5$, $\sigma_i = 1.0$, $\sigma_a = \pi$, $w_d = 0.4$, $w_i = 0.25$, $w_s = 0.15$, $w_a = 0.2$, and $\alpha = 0.75$.

2.3. Simulated data

We created three sequences of simulated image stacks to validate our particle tracking method. Every simulated series had 20 time points (dimensions 512 x 512 x 20, voxel size 0.207 x 0.207 x 0.500 μm^3, temporal resolution 30 seconds). For the first time point, particles were set to discrete locations with a random average intensity value between 96 and 128. The number and sizes of the particles were initialized using results of the real data analysis. The speed and direction of movement of each particle was set at every time point. Clustering of the particles was defined to increase their size and average intensity.

The first series was initialized with 103 particles, set to move in random directions at random speeds less than 34.5 nm/sec, without clustering. The second series was initialized with 154 particles, set to move at random speeds less than 69 nm/sec, with 15% of the particles closer than 1.4 μm to each other clustering at every time point. To facilitate clustering, 15% of the particles at every time point were set to move towards one of six randomly initialized target points. Other particles were set to move in random directions. The third series was set to have more merging of the particles than the second series. The speeds of the particles were set to be less than 138 nm/sec. 30% of the particles closer than 1.8 μm were set to cluster and 30% of the particles set to move towards three target points. All simulations were created with the BioImageXD software [9].

The particle tracking method was used to create trajectories for all three simulated series. Parameters were the same for all series, except for D_h: $D_l = 0.0\mu$m, $\sigma_d = 1.0$, $\sigma_s = 0.5$, $\sigma_i = 0.5$, $\sigma_a = \pi$, $w_d = 0.4$, $w_i = 0.25$, $w_s = 0.25$, $w_a = 0.1$, and $\alpha = 0.75$. D_h was set to 1.0μm, 1.5μm, or 2.5μm for the first, second, and third series respectively, to create trajectories with varying particle speeds.

3. RESULTS

3.1. Real data results

We analysed the movement of the particles in real data using four parameters: average number of time points in the tracks, average length of the tracks, average speed of the particles and average directional persistence. The average number of time points is defined as the duration of a track in frames. Directional persistence of track t is defined as

$$\frac{||(t_n)_l - (t_1)_l||}{\sum_{i=2}^{n} ||(t_i)_l - (t_{i-1})_l||}$$

where n is the number of time points in trajectory t.

The results (Table 1) show that average speed of the particles tends to slow down with time. Directional persistence seemed greatest in the first series, meaning less Brownian motion, although the difference is small. The average number of time points per track was smallest in the first series, possibly caused by more active merging of the particles. On the whole, the results of the five series were consistent, and in agreement with the biology of the integrin clustering phenomenon [10], indicating that our particle tracking method was properly measuring the phenomenon. An example of the trajectories created is shown in Figure 2.

Table 1. Results of the analysis of real microscopy data. Four parameters are shown for each series: average number of time points in tracks, average length of tracks, average speed of particles and average directional persistence.

Time point	Avg. tpts	Avg. length (μm)	Avg. speed (nm/sec)	Avg. dp
33 min	8.42	4.15 ± 0.16	21.8 ± 0.5	0.53 ± 0.02
62 min	10.37	3.75 ± 0.16	16.0 ± 0.5	0.44 ± 0.02
92 min	10.10	4.05 ± 0.12	19.6 ± 0.4	0.48 ± 0.01
118 min	9.57	3.76 ± 0.13	18.2 ± 0.5	0.48 ± 0.02
141 min	10.77	3.74 ± 0.18	15.4 ± 0.5	0.46 ± 0.02

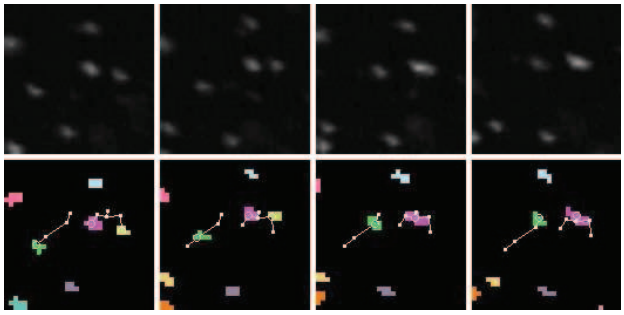


Fig. 2. Particle tracking results of real data of four consecutive time points. Upper row: small regions cut from a single plane of the original image data. Lower row: detected particles from the same regions, with three sample trajectories. Particles belonging to the same trajectory have the same color. Two of the trajectories merge in the third time point.

3.2. Simulated data results

We analysed the movement of the particles in simulated data using four parameters: number of tracks, average number of time points in tracks, average speed of particles and average directional persistence. The number of tracks was used to measure whether the particle tracking method correctly estimated merging of the particles.

We compared the ground truth of the simulated data (Table 2) to the particle tracking results of the same data (Table 3). The average speed and average directional persistence matched within error limits for all simulations. The number of tracks and the average number of time points in tracks also matched for simulation 1, but small discrepancies emerged with the other simulations, as expected with increasing particle speed. Still even the result for simulation 3 was reasonably good, considering that it is an extreme case, where the average speed of particles is over three times higher than the fastest movement detected from real data.

We compared our method to the previously published and validated method by Jaqaman et al [1]. We configured the Jaqaman method using the same values of parameters as used in our method, but since the methods have somewhat different sets of parameters, the results acquired with the Jaqaman method are only approxi-

Table 2. Ground truth of simulated data for method validation. For each simulation, four parameters are shown: number of tracks, average number of time points in tracks, average speed of particles in tracks and average directional persistence.

Series	# of tracks	Avg. tpts	Avg. speed (nm/sec)	Avg. dp
Simulation 1	103	20.0	17.4 ± 0.3	0.25 ± 0.02
Simulation 2	154	18.1	34.5 ± 0.5	0.30 ± 0.02
Simulation 3	146	16.2	69.3 ± 1.1	0.36 ± 0.02

Table 3. Results of the analysis of simulated data. For each analysed series, the same four parameters are shown as in Table 2.

Proposed method	# of tracks	Avg. tpts	Avg. speed (nm/sec)	Avg. dp
Simulation 1	104	19.8	17.9 ± 0.3	0.25 ± 0.02
Simulation 2	164	16.9	35.5 ± 0.6	0.32 ± 0.02
Simulation 3	177	13.3	68.5 ± 1.1	0.39 ± 0.02
Jaqaman method [1]	# of tracks	Avg. tpts	Avg. speed (nm/sec)	Avg. dp
Simulation 1	115	17.7	18.2 ± 0.5	0.28 ± 0.02
Simulation 2	173	15.7	33.8 ± 0.7	0.35 ± 0.02
Simulation 3	147	15.1	70.6 ± 1.7	0.35 ± 0.02

mate. Both methods gave fairly similar results for the average speed and the average directional persistence for all simulations, but our method generally produced better results for the number of tracks and the duration of tracks. Here the Jaqaman method performed better only with the extreme case of the fast moving and highly merging particles of simulation 3, most likely because of an additional linking step done to the tracks.

We also analysed the number of perfect tracks (tracks in the analysis results that have a complete match in the ground truth) and successful linking of individual particles between time points. The results show (Table 4) that our particle tracking method worked almost perfectly with simulation 1, which did not include any merging. Only one incorrect link was made due to a single merging of two particles, and as a consequence, two tracks were set as non-perfect and one new track was created. This illustrates the sensitivity of the perfect tracks metric. Simulation 2 gave a 98.9% accuracy for the links and over 50% of perfect tracks. The fast moving and highly clustering simulation 3 gave 26.0% of perfect tracks, but the linking accuracy was still high (94.7%), proving that the quantitative analysis of the movement of the particles can be reliably derived as presented in Table 3.

Our method resulted in a higher number of perfect tracks and correct links than the Jaqaman method, for all simulations. The results differed from the quantitative measurements in the fast moving and highly merging simulation 3. It seemed that longer tracks generated by the Jaqaman method often had at least one incorrect link assigned.

4. CONCLUSIONS

We have presented a new, greedy single-particle tracking method. This method can be used to efficiently track hundreds or thousands of particles, as it only needs one-pass through the particles in a time series. The method provides a solution to the particle correspondence problem, including handling of merging, splitting, appearing and disappearing events. Options to configure the method for numerous different applications were presented. We tested the method by quantifying particle motion in real, relevant microscopy data, with expected results. We then validated the accuracy of the method using simulated image series. Finally, we compared the accuracy of the method to a previously published, well known, single-particle tracking method. The results showed our tracking method to be reliable and functional.

Table 4. Results of the track analysis of simulated data. The number of perfect tracks compared to the total number of tracks, and the number of correct particle links compared to total particle links, are shown for each analysed series.

Proposed method	Tracks (perfect / total)	Links (correct / total)
Simulation 1	101 / 104 = 97.1%	1956 / 1957 = 99.9%
Simulation 2	94 / 164 = 57.3%	2579 / 2609 = 98.9%
Simulation 3	46 / 177 = 26.0%	2067 / 2183 = 94.7%
Jaqaman method [1]	Tracks (perfect / total)	Links (correct / total)
Simulation 1	79 / 115 = 68.7%	1883 / 1917 = 98.2%
Simulation 2	59 / 173 = 34.1%	2370 / 2535 = 93.5%
Simulation 3	4 / 147 = 2.7%	1594 / 2075 = 76.8%

5. REFERENCES

[1] K. Jaqaman et al., "Robust single-particle tracking in live-cell time-lapse sequences," *Nat. Methods*, vol. 5, no. 8, pp. 695–702, 2008.

[2] K. Shafique and M. Shah, "A noniterative greedy algorithm for multiframe point correspondence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, pp. 51–65, 2005.

[3] C.J. Veenman, M.J.T. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 54–72, 2001.

[4] E. Meijering, I. Smal, and G. Danuser, "Tracking in molecular bioimaging," *IEEE Signal Proc. Mag.*, vol. 23, no. 3, pp. 46–53, 2006.

[5] W. Tvarusko et al., "Time-resolved analysis and visualization of dynamic processes in living cells," *Proc. Natl. Acad. Sci. USA*, vol. 96, no. 14, pp. 7950–7955, 1999.

[6] P. Ruusuvuori et al., "Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images," *BMC Bioinformatics*, vol. 11, pp. 248, 2010.

[7] A. Dufour et al., "Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces," *IEEE Trans. Image Proc.*, vol. 14, no. 9, pp. 1396–1410, 2005.

[8] C. Zimmer et al., "On the digital trail of mobile cells," *IEEE Signal Proc. Mag.*, vol. 23, no. 3, pp. 54–62, 2006.

[9] P. Kankaanpää, L. Paavolainen, V. Marjomäki, J. Heino, and D. White, "BioImageXD," `http://www.bioimagexd.net/`, 2011.

[10] M. Karjalainen et al., "Echovirus 1 infection depends on biogenesis of novel multivesicular bodies," *Cell. Microbiol.*, vol. 13, no. 12, pp. 1975–1995, 2011.

[11] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.