Deep Particle Tracker: Automatic Tracking of Particles in Fluorescence Microscopy Images Using Deep Learning

Roman Spilger[1](✉), Thomas Wollmann[1], Yu Qiang[1], Andrea Imle[2],
Ji Young Lee[3], Barbara Müller[4], Oliver T. Fackler[2], Ralf Bartenschlager[3],
and Karl Rohr[1]

[1] Division of Bioinformatics, Biomedical Computer Vision Group,
University of Heidelberg, BioQuant, IPMB, and DKFZ Heidelberg,
Im Neuenheimer Feld 267, 69120 Heidelberg, Germany
roman.spilger@bioquant.uni-heidelberg.de

[2] Department of Infectious Diseases, Integrative Virology,
University Hospital Heidelberg, Heidelberg, Germany

[3] Department of Infectious Diseases, Molecular Virology,
University Hospital Heidelberg, Heidelberg, Germany

[4] Department of Infectious Diseases, Virology, University Hospital Heidelberg,
Heidelberg, Germany

Abstract. Tracking of particles in fluorescence microscopy image sequences is essential for studying the dynamics of subcellular structures and virus structures. We introduce a novel particle tracking approach using an LSTM-based neural network. Our approach determines assignment probabilities jointly across multiple detections by exploiting both short and long-term temporal dependencies of individual object dynamics. Manually labeled data is not required. We evaluated the performance of our approach using image data of the ISBI Particle Tracking Challenge as well as real fluorescence microscopy image sequences of virus structures. It turned out that the proposed approach outperforms previous methods.

1 Introduction

Tracking of multiple particles in time-lapse fluorescence microscopy image sequences is an important task to quantify the dynamic behavior of subcellular and virus structures. Since a large number of particles needs to be tracked to draw statistically sound conclusions, accurate and robust automatic tracking approaches are indispensable.

Previous work on tracking biological particles can be subdivided into deterministic and probabilistic methods. Deterministic approaches follow a two step-paradigm comprising particle localization and motion correspondence (e.g., [13,14]). Probabilistic approaches are formulated within a Bayesian framework and take into account uncertainties to improve the robustness. The solution

D. Stoyanov et al. (Eds.): DLMIA 2018/ML-CDS 2018, LNCS 11045, pp. 128–136, 2018.
https://doi.org/10.1007/978-3-030-00889-5_15

is determined using Kalman filters or particle filters (e.g., [1,2,4,9,11]). A disadvantage of traditional tracking methods is that a handcrafted similarity measure is used to determine the degree of correspondence between detections in successive images. In addition, a suitable dynamic model needs to be selected, and often tedious manual tuning of (numerous) parameters is required. Often, these approaches have difficulties in cluttered environments with clustering objects. Deep learning methods have the potential to improve the performance. This has been demonstrated for different tasks such as segmentation and classification in the fields of computer vision and medical image analysis (e.g., [5]), however, much less work exists on tracking.

In the field of computer vision, Milan et al. [10] proposed a recurrent neural network (RNN) for tracking pedestrians in video images of natural scenes. However, tracking pedestrians is quite different from tracking biological particles since the motion and shape are very different, and appearance is not a reliable cue. Also, in [10] a handcrafted similarity measure is used for correspondence finding. In addition, two separate networks need to be trained for state prediction and data association. Sadeghian et al. [12] introduced an appearance-based RNN for tracking pedestrians in video images. However, there the similarity measure for correspondence finding is determined independently for each detection, and information on missing detections is not provided by the network. Also, a fixed input sequence length is used (last 6 time points). For training, manually labeled data was used. Yao et al. [17] used a similar approach as in [12] to track microtubules in synthetic data. However, the similarity measure for correspondence finding is not jointly computed across multiple detections, and a fixed input sequence length is used (as in [12]). In addition, objects are not automatically detected but ground truth positions are used, and real microscopy data was not considered. He et al. [6] introduced an approach based on convolutional neural networks (CNNs) for tracking of cells. However, this approach does not use an RNN, and tracking of particles was not considered.

In this contribution, we introduce a new approach for particle tracking in time-lapse fluorescence microscopy images based on an RNN. Both short- and long-term temporal dependencies of individual object dynamics are exploited for state prediction and correspondence finding using a long short-term memory (LSTM) [7]. The network automatically learns to determine assignment probabilities for correspondence finding, without requiring a handcrafted similarity measure. In contrast to [12,17], our network computes assignment probabilities jointly across multiple detections, and also determines the probabilities of missing detections. In addition, the input sequence length is not limited but can be arbitrary long. Thus, we exploit more information and intrinsically cope with missing detections. Moreover our approach does not require manually labeled data (in contrast to [10,12,17]). Both state prediction and data association are trained within one network. Compared to traditional tracking methods, the dynamic model is automatically selected, and tuning of tracking parameters is not required. We performed a quantitative evaluation using data from the

ISBI Particle Tracking Challenge as well as using real live cell microscopy data of human immunodeficiency virus type 1 (HIV-1) particles and hepatitis C virus (HCV) proteins. It turned out that our approach yields better tracking results than previous methods.

2 Methods

Our approach, denoted as deep particle tracker (DPT), relies on a tracking-by-detection paradigm. For spot detection, we use the spot-enhancing filter (SEF) [13] yielding a set of detections. For correspondence finding, we introduce an LSTM-based recurrent neural network that determines assignment probabilities between tracked objects and particle detections. To establish one-to-one correspondences using the computed assignment probabilities of all objects and the probabilities of missing detections, the Hungarian algorithm is employed.

2.1 Network Architecture

In our DPT approach, for each object we use one neural network with the same network architecture. We employ both LSTM and fully-connected (FC) layers each consisting of K units (we used $K = 250$). We apply Gaussian dropout after each layer. Below, we describe the network architecture in more detail.

Let the vector $\mathbf{x}_t^i \in \mathrm{I\!R}^{\mathrm{D}}$ denote the state of an object i at time point t. In our work, we used $\mathbf{x}_t^i = (x_t^i, y_t^i, s_t^i, \alpha_t^i)$, i.e. $D = 4$. (x_t^i, y_t^i) is the object position. The speed and direction of the object motion is denoted by s_t^i and α_t^i (computed using the positions at two successive time points). The detections (positions as well as speed and direction for an assignment to object i) are represented by the vector $\mathbf{y}_t^i \in \mathrm{I\!R}^{\mathrm{M \cdot D}}$, where M is the overall number of detections. Note that M is often very high (in cluttered environments) and varies strongly between different images of a sequence. On the other hand, the neural network requires a fixed input vector size. To address this, in our approach we exploit the M-nearest detections (we used $M = 5$). For each time point $t - 1$, the network computes two output vectors for the next time point t: $\hat{\mathbf{x}}_t^i \in \mathrm{I\!R}^{\mathrm{D}}$ is the predicted object state, and $\mathbf{a}_t^i \in [0, 1]^{M+1}$ represents the assignment probabilities between object i and the M-nearest detections as well as probabilities for missing detections.

We use an LSTM to predict the state of an object i for the next time point t. The LSTM is composed of layers interacting which each other to determine the new hidden state $\mathbf{h}_t^i \in \mathrm{I\!R}^{\mathrm{K}}$ of dimension K which also represents the output. The main component of an LSTM is the cell state $\mathbf{c}_t^i \in \mathrm{I\!R}^{\mathrm{K}}$ which serves as long-term memory [7]. At each time point t, different types of gates determine which information is added to or removed from the previous cell state $\mathbf{c}_{t-1}^i$. Note that all gates compute their output based on the previous hidden state $\mathbf{h}_{t-1}^i$ and the current input. In our case, the input is the object state $\mathbf{x}_{t-1}^i$ mapped to the vector $\mathbf{z}_t^i \in \mathrm{I\!R}^{\mathrm{K}}$ by using a fully-connected (FC) layer and a hyperbolic tangent activation function. At time point t, the LSTM for an object i is updated as follows:

$$\mathbf{i}_t^i = \sigma(\mathbf{W}_{zi}\mathbf{z}_t^i + \mathbf{W}_{hi}\mathbf{h}_{t-1}^i + \mathbf{b}_i) \tag{1}$$

$$\mathbf{f}_t^i = \sigma(\mathbf{W}_{zf}\mathbf{z}_t^i + \mathbf{W}_{hf}\mathbf{h}_{t-1}^i + \mathbf{b}_f) \tag{2}$$

$$\mathbf{o}_t^i = \sigma(\mathbf{W}_{zo}\mathbf{z}_t^i + \mathbf{W}_{ho}\mathbf{h}_{t-1}^i + \mathbf{b}_o) \tag{3}$$

$$\mathbf{g}_t^i = \tanh(\mathbf{W}_{zg}\mathbf{z}_t^i + \mathbf{W}_{hg}\mathbf{h}_{t-1}^i + \mathbf{b}_g) \tag{4}$$

$$\mathbf{c}_t^i = \mathbf{f}_t^i \otimes \mathbf{c}_{t-1}^i + \mathbf{i}_t^i \otimes \mathbf{g}_t^i \tag{5}$$

$$\mathbf{h}_t^i = \mathbf{o}_t^i \otimes \tanh(\mathbf{c}_t^i) \tag{6}$$

where $\mathbf{i}_t^i$ is the input gate, $\mathbf{f}_t^i$ is the forget gate, $\mathbf{o}_t^i$ is the output gate, and $\mathbf{g}_t^i$ is the input modulation gate. Weight matrices $\mathbf{W} \in \mathrm{I\!R}^{\mathrm{K}\times\mathrm{K}}$ and bias vectors $\mathbf{b} \in \mathrm{I\!R}^{\mathrm{K}}$ represent the parameters of a gate. σ is the logistic sigmoid activation function, and $\otimes$ denotes element-wise multiplication. We use the new hidden state $\mathbf{h}_t^i$ of the LSTM to compute the predicted object state $\hat{\mathbf{x}}_t^i$ by employing a FC layer and a hyperbolic tangent activation function. Since $\mathbf{h}_t^i$ is a function of all object states $\mathbf{x}_{1:t-1}^i$ from time point 1 to time point $t-1$, the network can exploit both short and long-term temporal dependencies for state prediction.

The vector $\mathbf{y}_t^i$ of the detections is passed to a FC layer with a hyperbolic tangent activation function for mapping it to a K-dimensional vector, which is then concatenated with the hidden state $\mathbf{h}_t^i$ of the LSTM. The resulting vector of dimension $2K$ is passed to another FC layer which maps it to a vector of dimension K. This vector is fed into a fully connected linear output layer with softmax normalization so that the final network output vector $\mathbf{a}_t^i$ can be interpreted as $M+1$ assignment probabilities, i.e. $\forall i : \sum_{j=1}^{M+1} a_t^{ij} = 1$, where a_t^{ij} denote the assignment probabilities between object i and detection j ($j = 1, ..., M$), and $a_t^{i(M+1)}$ are the probabilities of missing detections. The computed assignment probabilities and the probabilities for missing detections (dummy detections in the probability matrix) are used as input for the Hungarian algorithm. Note that a handcrafted similarity measure for the predicted state and the detections (e.g., Euclidean distance) is not required to compute the assignment probabilities.

The LSTM-based neural network is trained by minimizing the loss $\mathcal{L}$ over all trajectories defined by:

$$\mathcal{L} = \sum_{i=1}^{N} \mathcal{L}^i, \qquad \mathcal{L}^i = \sum_{t=1}^{T^i} \left(\frac{1}{D} \|\hat{\mathbf{x}}_t^i - \tilde{\mathbf{x}}_t^i\|^2 - \sum_{j=1}^{M+1} \tilde{a}_t^{ij} \log(a_t^{ij}) \right) \tag{7}$$

where N is the overall number of trajectories, $\mathcal{L}^i$ denotes the loss for the trajectory of object i, $\hat{\mathbf{x}}_t^i$ is the predicted state and $\tilde{\mathbf{x}}_t^i$ the true state at time point t. The deviation between the states is quantified by the mean squared error (MSE). The cross-entropy is used to measure the deviation between the computed assignment probabilities a_t^{ij} and the ground truth $\tilde{a}_t^{ij}$. T^i defines the total number of time points for a trajectory.

2.2 Training

Since deep learning architectures involve a large number of parameters, vast amounts of training data are generally required. However, ground truth for microscopy image sequences of biological particles is hardly available and manual annotation is very tedious. Therefore, in our approach we do not use manually labeled data but rely on synthetic data for training. We generated a large number of simulated trajectories of particles, which perform Brownian motion or directed motion. The diffusion coefficients and velocities of individual particles were sampled from a uniform distribution and the initial positions were chosen randomly. In addition, we randomly removed particle positions which enables the network to learn coping with missing detections.

For training our network, we used the RMSprop optimizer [15] with an initial learning rate of 3×10^{-5}, which was decreased by 5% when the validation loss stopped improving. To avoid overfitting, we employed early stopping and set the Gaussian dropout rate to $p = 0.2$. We used a dataset with 85,000 synthetically generated trajectories with variable track length. The dataset was split into 82% for training and 18% for validation. We used a mini-batch size of 10 trajectories.

3 Experimental Results

3.1 Particle Tracking Challenge Data

We evaluated our DPT approach based on data of the ISBI Particle Tracking Challenge [2] and compared the performance with the overall top-three approaches (Methods 5, 1, and 2). Method 5 uses the spot-enhancing filter (SEF) [13] for particle localization and probabilistic data association [4]. Method 1 employs intensity-weighted centroids for particle localization and combinatorial optimization [14]. Method 2 localizes particles by local maxima selection and performs linking by multiple hypothesis tracking [3]. In addition, we compared the performance of DPT with a recent approach employing a piecewise-stationary motion model smoother (PMMS) [11]. This approach uses SEF for particle localization and linear programming for linking (extension of u-track [8]).

To study the performance in cluttered environments, we used data of the vesicle scenario for signal-to-noise ratios of SNR = 4 and SNR = 7 as well as medium and high particle densities (medium: 500 particles/frame, high: 1000 particles/frame). The data is challenging due to conflicting correspondences (in total 15,682 trajectories). The image sequences consist of 100 images (512×512 pixels) with random appearance and disappearance of particles. To quantitatively assess the performance of the tracking methods, we computed the metrics α, β, JSC, JSC_θ, and $RMSE$ as described in [2]. $\alpha \in [0, 1]$ indicates the overall degree of matching of ground truth and estimated tracks excluding spurious tracks. $\beta \in [0, \alpha]$ includes an additional penalization for spurious tracks compared to α. The Jaccard similarity coefficient $JSC \in [0, 1]$ represents the overall particle detection performance, and $JSC_\theta \in [0, 1]$ is the rate of correctly estimated tracks. The overall localization accuracy is indicated by the root mean square error ($RMSE$).

The quantitative results are presented in Table 1 (bold values indicate the best performance). It can be seen that DPT performs best for all metrics and cases. Note that for PMMS the results in [11] are given only up to two decimal places and $RMSE$ is not provided. Note that for our DPT approach, we did not use the Particle Tracking Challenge data for training, but used our own generated synthetic data as described in Sect. 2.2 above.

Table 1. Tracking performance of different approaches for data of the vesicle scenario from the Particle Tracking Challenge. Bold indicates best performance.

Density	Meth	SNR = 4					SNR = 7				
		α	β	JSC	JSC_θ	$RMSE$	α	β	JSC	JSC_θ	$RMSE$
Med	Meth 5	0.658	0.588	0.641	0.776	0.754	0.677	0.605	0.646	0.783	0.667
	Meth 1	0.687	0.609	0.652	0.767	0.607	0.700	0.619	0.650	0.758	0.544
	Meth 2	0.582	0.514	0.590	0.757	0.970	0.611	0.547	0.606	0.775	0.828
	PMMS	0.67	0.60	0.64	0.77	-	0.68	0.61	0.64	0.78	-
	DPT	**0.695**	**0.624**	**0.658**	**0.790**	**0.545**	**0.711**	**0.631**	**0.651**	**0.790**	**0.525**
High	Meth 5	0.488	0.408	0.466	0.671	1.004	0.533	0.453	0.503	0.698	0.931
	Meth 1	0.531	0.442	0.487	0.641	0.801	0.582	0.494	0.526	0.683	0.683
	Meth 2	0.430	0.356	0.429	0.649	1.208	0.466	0.395	0.458	0.665	1.027
	PMMS	0.51	0.44	0.48	0.67	-	0.55	0.48	0.51	0.69	-
	DPT	**0.547**	**0.462**	**0.505**	**0.680**	**0.746**	**0.590**	**0.507**	**0.535**	**0.702**	**0.677**

3.2 Real Fluorescence Microscopy Images of Virus Structures

We also evaluated the performance of the DPT approach using real fluorescence microscopy image sequences displaying human immunodeficiency virus type 1 (HIV-1) particles and hepatitis C virus (HCV) proteins. The fluorescence labeled HIV-1 particles were imaged by a confocal spinning disk microscope and an EM-CCD camera. For our evaluation we used two image sequences (each 50 time points, 1000 × 1000 pixels, 16-bit) denoted by Seq. A and Seq. B. We also used one image sequence showing the HCV nonstructural protein 5A (30 time points, 1000 × 1000 pixels, 16-bit) denoted by Seq. C (an example section with 115 × 115 pixels is shown in Fig. 1). The images were acquired by a confocal spinning disk microscope and a CMOS camera. This dataset is challenging due to relatively low SNRs and clutter (high particle density, often crossing of trajectories). Ground truth trajectories for regions with clutter and large motion were determined by manual annotation. Seq. A, Seq. B, and Seq. C comprise 117, 125, and 55 ground truth trajectories, respectively (with up to 30 time points).

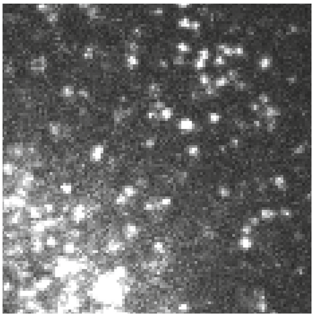
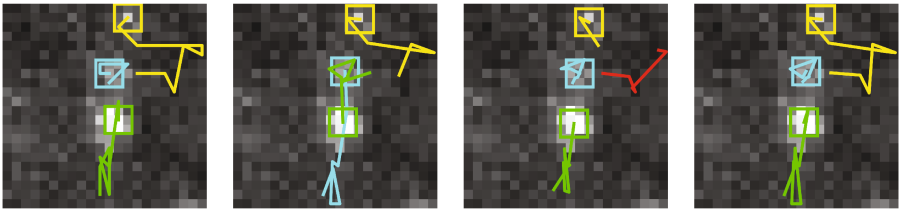


Fig. 1. Section of image sequence Seq. C (HCV). The image contrast was enhanced.

Table 2. Tracking performance of different approaches for real fluorescence microscopy images. Bold indicates best performance.

Sequence	Meth	α	β	JSC	JSC_θ	$RMSE$
Seq. A (HIV-1)	PT	0.312	0.255	0.348	0.442	2.701
	KF	0.388	0.317	0.421	0.456	2.775
	MHT	0.367	0.304	0.454	0.440	3.393
	DPT	**0.413**	**0.360**	**0.462**	**0.497**	**2.673**
Seq. B (HIV-1)	PT	0.328	0.261	0.338	0.399	2.559
	KF	0.352	0.312	0.396	0.373	**2.121**
	MHT	0.366	0.303	0.429	0.416	2.991
	DPT	**0.435**	**0.331**	**0.444**	**0.527**	2.717
Seq. C (HCV)	PT	0.590	0.496	0.629	0.557	1.064
	KF	0.559	0.481	0.564	0.550	1.088
	MHT	0.540	0.480	0.588	0.611	1.237
	DPT	**0.647**	**0.571**	**0.669**	**0.625**	**1.024**

We compared the performance of DPT with the ParticleTracker (PT) [14], a Kalman filter based approach (KF) [16], and multiple-hypothesis tracking (MHT) using multiple motion models [1]. PT uses intensity-weighted centroids for particle localization and combinatorial optimization [14]. KF uses SEF for particle localization and particle linking is based on a linear assignment method used in u-track [8]. MHT employs a wavelet-based detection scheme for particle localization. For PT, KF, and MHT we performed a grid search to determine optimal parameter settings. Note that for DPT, adaption of tracking parameters was not necessary (except the two detection parameters for SEF), i.e. we directly applied our tracking approach to the real data while training was performed only on synthetic data (see Sect. 2.2 above). Table 2 shows the tracking performance for all three image sequences. It turns out that DPT outperforms the other methods for all metrics and sequences (except $RMSE$ for Seq. B). Sample results for Seq. C are shown in Fig. 2. It can be seen that DPT yields the best result and maintains the correct identity for all three particles. KF causes an identity switch (between the blue and green trajectory). MHT yields a broken trajectory (yellow).



(a) Ground truth (b) KF (c) MHT (d) DPT

Fig. 2. Ground truth and results of different tracking approaches for image sequence Seq. C (HCV). The image contrast was enhanced for better visualization. (Color figure online)

4 Conclusion

We presented a novel approach for tracking particles in time-lapse microscopy images using an LSTM-based recurrent neural network which computes assignment probabilities jointly across multiple detections and also determines probabilities for missing detections. Manually labeled data is not required. In addition, a handcrafted similarity measure is not needed. We evaluated our approach based on synthetic and real image sequences. It turned out that our approach yields better results than previous methods.

Acknowledgment. Support of the DFG (German Research Foundation) within the SFB 1129 is gratefully acknowledged.

References

1. Chenouard, N., Bloch, I., Olivo-Marin, J.C.: Multiple hypothesis tracking for cluttered biological image sequences. IEEE Trans. Pattern Anal. Mach. Intell. **35**(11), 2736–3750 (2013)
2. Chenouard, N., et al.: Objective comparison of particle tracking methods. Nat. Methods **11**(3), 281–289 (2014)
3. Coraluppi, S., Carthel, C.: Multi-stage multiple-hypothesis tracking. J. Adv. Inf. Fusion **6**, 57–67 (2011)
4. Godinez, W.J., Rohr, K.: Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. IEEE Trans. Med. Imag. **34**(2), 415–432 (2015)
5. Greenspan, H., van Ginneken, B., Summers, R.M.: Deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans. Med. Imag. **35**(5), 1153–1159 (2016)
6. He, T., Mao, H., Guo, J., Yi, Z.: Cell tracking using deep neural networks with multi-task learning. Image Vis. Comput. **60**, 142–153 (2017)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
8. Jaqaman, K., et al.: Robust single-particle tracking in live-cell time-lapse sequences. Nat. Methods **5**(8), 695–702 (2008)
9. Liang, L., Shen, H., Camilli, P.D., Duncan, J.S.: A novel multiple hypothesis based particle tracking method for clathrin mediated endocytosis analysis using fluorescence microscopy. IEEE Trans. Image Process. **23**(4), 1844–1857 (2014)
10. Milan, A., Rezatofighi, S.H., Dick, A., Reid, I., Schindler, K.: Online multi-target tracking using recurrent neural networks. In: Proceedings of 2017 Conference on Artificial Intelligence (AAAI), San Francisco, CA, USA, pp. 4225–4232, February 2017
11. Roudot, P., Ding, L., Jaqaman, K., Kervrann, C., Danuser, G.: Piecewise-stationary motion modeling and iterative smoothing to track heterogeneous particle motions in dense environments. IEEE Trans. Image Process. **26**(11), 5395–5410 (2017)
12. Sadeghian, A., Alahi, A., Savarese, S.: Tracking the untrackable: learning to track multiple cues with long-term dependencies. In: Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 300–311, October 2017

13. Sage, D., Neumann, F.R., Hediger, F., Gasser, S.M., Unser, M.: Automatic tracking of individual fluorescence particles: application to the study of chromosome dynamics. IEEE Trans. Image Process. **14**(9), 1372–1383 (2005)
14. Sbalzarini, I., Koumoutsakos, P.: Feature point tracking and trajectory analysis for video imaging in cell biology. J. Struct. Biol. **151**(2), 182–195 (2005)
15. Tieleman, T., Hinton, G.: Lecture 6.5-RMSPROP: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw. Mach. Learn. **4**(2), 26–31 (2012)
16. Tinevez, J.Y., et al.: TrackMate: an open and extensible platform for single-particle tracking. Methods **115**, 80–90 (2017)
17. Yao, Y., Smal, I., Meijering, E.: Deep neural networks for data association in particle tracking. In: Proceedings of 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, D.C., USA, pp. 458–461, April 2018