Online Neural Cell Tracking using Blob-Seed Segmentation and Optical Flow

Jingru Yi[1] Pengxiang Wu[1] Qiaoying Huang[1] Hui Qu[1]
Daniel J. Hoeppner[2] Dimitris N. Metaxas[1]
[1] Department of Computer Science, Rutgers University, NJ 08854, USA
[2] Lieber Institute for Brain Development, MD 21205, USA
{jy486, pw241, qh55, hq43}@cs.rutgers.edu
daniel.hoeppner@astellas.com dnm@cs.rutgers.edu

Abstract

Existing neural cell tracking methods generally use the morphology cell features for data association. However, these features are limited to the quality of cell segmentation and are prone to errors for mitosis determination. To overcome these issues, in this work we propose an online multi-object tracking method that leverages both cell appearance and motion features for data association. In particular, we propose a supervised blob-seed network (BSNet) to predict the cell appearance features and an unsupervised optical flow network (UnFlowNet) for capturing the cell motions. The data association is then solved using the Hungarian algorithm. Experimental evaluation shows that our approach achieves better performance than existing neural cell tracking methods.

1. Introduction

One primary aim of stem cell biology is to understand the factors influencing cell fate when multipotent cells become specified as terminal functional cell types. In the central nervous system, neural stem cells become specified as neurons, astrocytes, and oligodendrocytes. During this process, cells are constantly sampling their environment, making transient and long-term contacts with neighboring cells via filopodia and lamellipodia. Such behavior is typically recorded as time-lapse videos where the vision techniques could be applied for automatic analysis. In particular, cell tracking serves as an essential tool for the study of cell-cell interactions and thereby plays an important role in discovering the factors that influence the specified fate, and more importantly, the specific changes that correlate with disease progression [2].

Visual tracking for multiple targets in videos has been widely studied for many years. This task aims to find the optimal set of trajectories for moving objects within a video. In cell tracking, particularly neural cell tracking problems, one widely adopted strategy is tracking-by-segmentation [38], where cells are segmented and then associated over frames. However, existing methods [35, 2, 30, 26] typically suffer from inaccurate segmentation due to the complexities of neural cell images, which involve tiny cell structures, unclear cell boundaries, cell adhesion and background impurities. Moreover, their data association metrics are too simple and limited to hand-crafted cell morphology attributes. As a result, they fail to build association accurately and produce large tracking errors.

Recent advances in deep learning have brought revolutionary developments in visual tracking techniques. For multi-object tracking, a prevalent methodology is tracking-by-detection [6, 34, 25, 20], where the targets are localized first, and then associated between frames. In data association, metrics such as bounding box intersection-over-union (IoU), spatial-temporal distance and appearance similarity are used to measure how likely two targets belong to the same identity. In particular, appearance similarity serves as one of the most critical metrics; however, it tends to result in errors for similar targets. As a complement, optical flow-based motion information is commonly incorporated for more robust tracking [9, 39, 25]. However, the standard optical flow methods [7, 12] exploited in their works are computationally expensive, and their parameters need to be manually adjusted for different situations. Such weakness is recently overcome by training end-to-end optical flow convolutional neural networks (CNN) on synthetic datasets [11, 17]. However, due to the limited variability of synthetic datasets, as well as the domain difference between generated data and real-world imagery, the generalization of these methods remains challenging. To alleviate this problem, unsupervised optical flow networks [18, 32, 37, 27] are developed, yet with another shortcoming that their performance is limited by occlusion and large motion [37].

In this paper, we introduce an online tracking-by-segmentation method (see Fig. 1) that combines both ap-

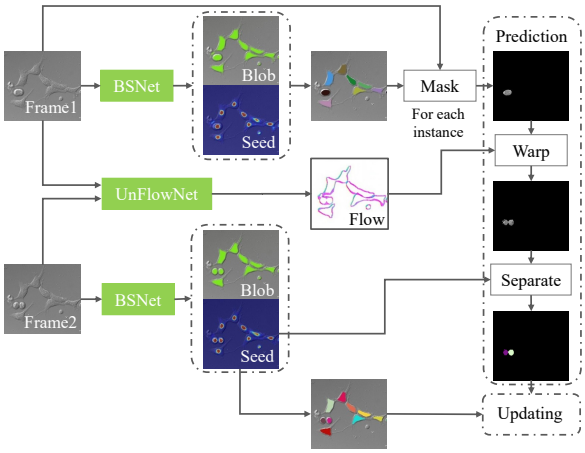

Figure 1. Illustration of our online tracking-by-segmentation process. The images are cropped from original ones. We show a mitosis data association example here. The updating process is based on mask IoU and is performed with Hungarian algorithm [23, 28].

pearance and motion features for neural cell tracking. Our method consists of two components (see Fig. 2): a supervised blob-seed network (BSNet) and an unsupervised optical flow network (UnFlowNet). BSNet provides cell instance appearance features, while UnFlowNet captures cell motions between frames. The online tracking (see Fig. 1) comprises two processes: mask prediction and updating. In the updating process, we employ the Hungarian algorithm [23, 28] for data association and adopt mask intersection-over-union (IoU) as the association metric. We verify the proposed method through a series of experiments, which demonstrate its superiority in neural cell tracking.

2. Related Work

2.1. Multi-object Tracking

One prevalent paradigm for multi-object tracking is tracking-by-detection. The general idea is to first localize the targets using an object detector in each frame, and then associate the targets across frames. One typical class of instances is the online methods [8, 39, 21, 6], which associate detections of the incoming frame immediately to existing trajectories and are efficient for real-time scenarios. Trajectories are typically handled with state-space models such as Kalman [19] and particle filters [13]; and the bounding box association for each tracker-detection pair aims to minimize defined assignment cost, which can be solved via Hungarian algorithm [23, 28] or greedy association [8]. Our method also works in an online manner.

Different from online approaches, the offline methods typically construct a set of trajectories through global or delayed optimization. For example, network flow-based methods [3, 44, 31] model the problem as a graph which can be solved globally and efficiently. However, these methods are very restrictive in representing motion and appearance due to the intrinsic properties of cost functions. Inspired by the success of deep learning, Kim *et al*. [20] propose to incorporate deep appearance features into multiple hypotheses tracking to solve the multidimensional assignment problem. Similarly, Son *et al*. [34] learn the cost metric through deep network, while the Siamese networks [25, 5], triplet network [16] and correlation network [36] learn to discriminate whether the two input image patches belong to the same trajectory or not. These deep feature based tracking methods typically require well-defined supervision for the learning of appropriate data association metric.

To make data association more reliable, standard optical flow methods [7, 12] are utilized to incorporate motion information in multi-object tracking [9, 39, 25]. To accelerate the computation and avoid manual parameter tuning of optical flow, deep learning based methods are developed. Representative examples include FlowNet [11] and FlowNet 2.0 [17]. Training these deep models typically requires a large amount of data, and is thus commonly performed on synthetic dataset. However, due to domain difference, it is difficult to apply the trained model directly to real-world imagery such as microscopy images. To solve this issue, Jason *et al*. [18, 32] propose unsupervised optical flow networks where the original supervised loss is replaced by a proxy loss based on the classical brightness constancy and smoothness assumptions. Wang *et al*. [37, 27] further design bidirectional flow to alleviate the occlusion problem. In this paper, we employ the forward unsupervised flow to capture cell motions. However, different from existing works, we avoid the influence of background and large motions by masking the images with detected cell blobs, thereby largely improving the flow accuracy.

2.2. Neural cell tracking

Existing neural cell tracking methods generally adopt tracking-by-segmentation methodology. For example, Tang *et al*. [35] use fuzzy threshold, watershed and geometric snakes to segment cells across the whole sequences. Al-Kofahi *et al*. [2, 30, 26] use seeded watershed to overcome the segmentation errors caused by touching cells. These unsupervised hand-crafted methods are sensitive to image intensity variations and suffer from over- and under- segmentation. To track the cells, Pinidiyaarachchi *et al*. [30] propagate the seeds and tracker identities from the previous frame to the current frame. However, the propagation fails to identify cell mitosis and the tracking performance depends heavily on segmentation qualities. Al-Kofahi *et al*. [2, 35, 26] model the probabilities of cell movement and division using cell morphology attributes, such as centroid, the major axis, and orientation. These prediction models tend to generate large tracking errors due to inac-

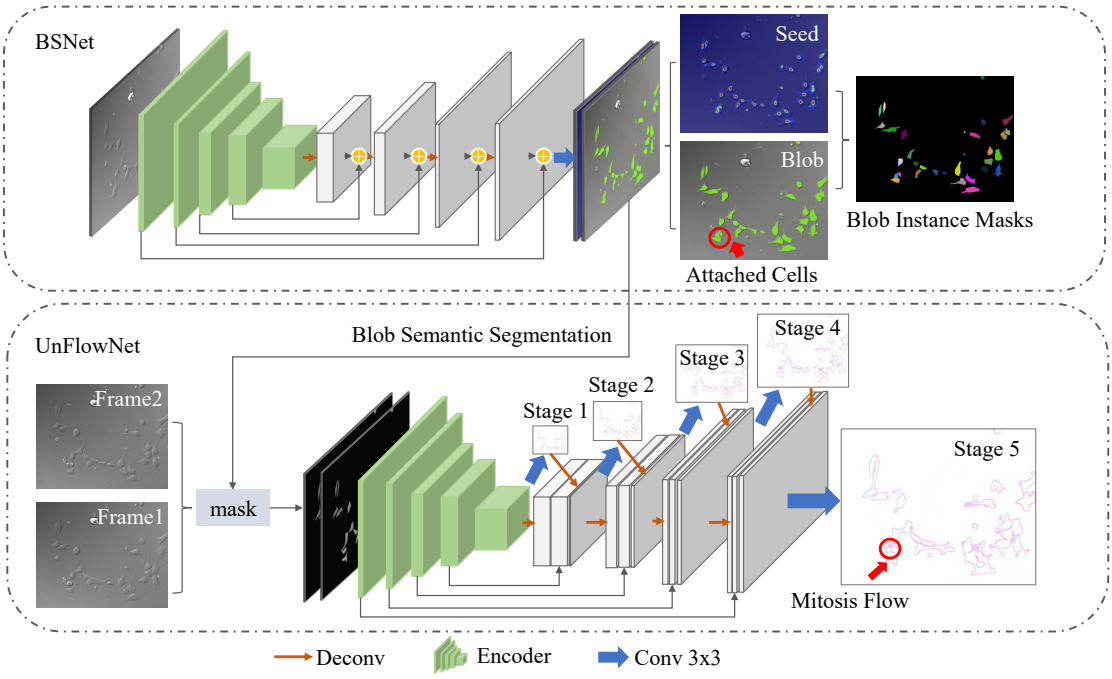

Figure 2. The two architectures: BSNet and UnFlowNet. The encoder is composed of conv0, conv1, conv2_x, con3_x, conv4_x from left to right, where structures of conv1-conv4 are from ResNet50 [15] and conv0 contains two 3×3 convolutional layers. The red circle in BSNet indicates the attached cells that can hardly be separated by blob map alone. The red circle in the UnFlowNet points to the learned optical flow that can be used to predict cell mitosis.

curate segmentation as well as abnormal cell behaviors and shapes. Compared to existing works above, our method is able to identify cell mitosis and produce accurate blob masks, thanks to the learned deep features.

3. Method

Our online tracking-by-segmentation method (see Fig. 1) comprises two components: a supervised blob-seed network (BSNet) and an unsupervised optical flow network (UnFlowNet). The BSNet provides the instance blob segmentation for each cell and the UnFlowNet captures the cell motions between two frames for mask prediction. We first introduce the two components in section 3.1 and section 3.2. Then we demonstrate our tracking algorithm in section 3.3.

3.1. BSNet

For neural cell tracking, one key problem is to identify the neural cell instances. However, neural cells tend to contact each other transiently or long-termly, making it difficult to distinguish different instances. Instance segmentation is one possible solution to this problem [41, 14, 42, 43], while it suffers from a huge imbalance between positive and negative anchor boxes [24, 40]. In response to such issues, we propose to use blob segmentation to represent the cell appearance features for neural cell tracking. The blobs of cells we utilize here are the areas inside the contours of cells, where the filopodia and lamellipodia are not included. Cell blobs are effective to tell apart different targets, but could be insufficient for separating attached cells (see Fig. 1 and Fig. 2). To deal with this problem, we further employ seed heatmaps to help separate the cells. In particular, we detect the number of seeds inside the blob area. When the number of seeds is greater than one, we apply watershed to separate the connected blobs.

We develop a supervised blob-seed network (BSNet) for the prediction of cell seeds and blobs. The seed heatmaps and the blob segmentations are combined to capture the appearance features of cell instances, which will be further used for mask prediction and mask association. As shown in Fig. 2, the BSNet structure is similar to a U-Net [33]. The encoder contains five convolutional layer blocks, which we name them from left to right conv0, conv1, conv2_x, con3_x, and conv4_x. The structures of conv1-conv4 are from ResNet50 [15]. Conv0 consists of two 3×3 convolutional layers. We use one 4×4 deconvolutional layers with stride 2 in each skip connection with a plus operator to combine the shallow and deep features. The head prediction of BSNet is a two-channel feature map containing

seed heatmap and blob map of the neural cells. The blob and seed segmentations are then combined to generate the blob instance masks. We normalize the two output maps by a pixelwise sigmoid function. The objective loss of the BSNet is a binary cross-entropy function.

3.2. UnFlowNet

Data association is one of the most critical problems for cell tracking. Existing cell tracking methods generally use empirical morphology metrics to associate cell instances. To detect mitosis during tracking, overlaps [1], distances [2], and morphology combinations [10] between parents and children are employed. However, these metrics are ineffective in scenarios where there exist touching cells and morphology changes.

In this work, we propose to utilize the optical flow field to correlate the cell instances between two frames directly (see Fig. 1). In particular, for each cell blob instance in the previous frame, we warp it to the current frame using the optical flow field. Then the instance association between the predicted blobs and the blob segments of the current frame is solved by Hungarian algorithm [23, 28] using mask IoU. Our method has two major advantages. First, the cell mitosis can be captured by optical flow (see Fig. 1 and Fig. 2). Compared to existing complex restricted morphology metrics, our warping-based mitosis prediction is more reliable. Second, the optical flow field predicts the possible morphology and position changes of previous cell instances in the current frame, thereby providing more accurate instance association.

We employ an unsupervised optical flow network (UnFlowNet) to learn the motions between two consecutive frames. To avoid the interference of background change and large cell motion caused by filopodia and lamellipodia, we mask the input images with the blob segmentations generated from BSNet. As shown in Fig. 2, the UnFlowNet contains the same encoder architecture as BSNet. The coarse-to-fine optical flow refinement parts follow the designs of FlowNetS [11]. We use the smooth and photometric loss [18] to optimize the network parameters. In particular, we calculate the losses between the input image pairs and output optical flow at every stage ($s = 1, 2, \ldots, 5$). For input images $\mathbf{I}_t, \mathbf{I}_{t+1} \in \mathbb{R}^{3\times H\times W}$, the loss function is defined as:

$$\mathcal{L} = \sum_{s=1}^{5} (\mathcal{L}_{\text{photometric}}(\mathbf{u}_s, \mathbf{v}_s, \mathbf{I}_t, \mathbf{I}_{t+1}) + \mathcal{L}_{\text{smooth}}(\mathbf{u}_s, \mathbf{v}_s)), \tag{1}$$

where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{H\times W}$ represent the horizontal and the vertical flow between the two input images and s indexes the stage number of output optical flow (see Fig. 2). The photometric loss is the sum of difference between $\mathbf{I}_{t+1}$ and the warped images from $\mathbf{I}_t$:

$$\mathcal{L}_{\text{photometric}}(\mathbf{u}, \mathbf{v}, \mathbf{I}_t, \mathbf{I}_{t+1}) = \sum_{i,j} \rho(\mathbf{I}_{t+1}(i,j) - \mathbf{I}_t(i + u_{i,j}, j + v_{i,j})), \tag{2}$$

where $\rho(x) = (x^2 + \epsilon^2)^\alpha$ is the Charbonnier penalty [18]. The smooth loss is as follows:

$$\mathcal{L}_{\text{smooth}}(\mathbf{u}, \mathbf{v}) = \sum_{i,j} (\rho(u_{i,j} - u_{i+1,j}) + \rho(u_{i,j} - u_{i,j+1}) + \rho(v_{i,j} - v_{i+1,j}) + \rho(v_{i,j} - v_{i,j+1})). \tag{3}$$

3.3. Tracking Algorithm

Our online tracking-by-segmentation method for neural cell tracking is illustrated in Algorithm 1. For two sequential images x_1 and x_2, BSNet is first applied to frame x_1 to obtain its cell blob instance masks. We use N to represent the total number of masks. Next, we assign a tracker ID t_i to the i-th mask. We use $T = \{t_i\}_{i=1}^{N}$ to represent the tracker sets for frame x_1. Algorithm 1 is then used to predict and update the tracker sets. Note that each tracker t_i is associated with a cell blob mask.

Prediction Given the two sequential images x_1 and x_2 and the tracker sets T for x_1, the prediction process is to predict cell blob instance masks for x_2 and update the tracker set T. First, we obtain the optical flow field f between frame x_1 and x_2 using UnFLowNet (Line 1). For each t_i in T, we apply the Hadamard product between the instance mask of t_i and x_1 to mask out the interference of other cells, background and filopodia-like structures. And then we warp the product image using optical flow field f (Line 5). We denote the warped image by y. To check if y contains mitosis, we count the seeds of y using the seed heatmap from x_2 and separate y when the seeds number is 2 (see Fig. 1). If no mitosis happens, we replace the mask of t_i with y (Line 6). Otherwise, we use an adding set M^a and a removing set M^r to store the mitosis predictions. If a mitosis happens to y, we put its parent tracker t_i to M^r and store the two child masks in M^a. Our aim is to remove the tracker ID of the parent and create new tracker IDs for child cells. After iterating all the tracker t_i in T, as in Line 9, we remove t_i from T if t_i is in M^r and we create new tracker IDs for masks in M^a and add the new trackers to T. The two sets M^a and M^r are then emptied.

Updating After the previous step, our updated tracker set T contains the predicted blob masks for x_2 which ideally are consistent with the observed blob instance masks for x_2 in terms of shape, size and the number of instances. Our next step is to associate the predicted masks with the observed masks using the Hungarian algorithm [23, 28]. First,

we apply BSNet to x_2 to obtain its blob cell instance masks B, which are referred as the observed instance masks for x_2 (Line 2). Then we find the matched pairs between B and T using the Hungarian algorithm based on the mask IoU between each pair (Line 10). The matched pair indices are denoted by M. We update the instance mask of t_i with b_j if (i, j) is in M (Line 12). We add b_j to M^a and push t_i into M^r if (i, j) is not in M (Line 13). Finally, we remove t_i from T if t_i is in M^r. We create new tracker IDs for masks in M^a and add the new trackers to T (Line 15).

Algorithm 1: Online tracker prediction and updating

Input: frame x_1 and x_2, tracker $T = \{t_i\}_{i=1}^{N}$ of x_1
Output: updated tracker T for x_2

1 $f \leftarrow$ UnFlowNet(x_1, x_2) ; ▷ f : optical flow field
2 $B = \{b_j\}_{j=1}^{M} \leftarrow$ BSNet(x_2) ; ▷ B : blob mask instances
3 $M^a \leftarrow \emptyset$; $M^r \leftarrow \emptyset$;
4 **for** $i \leftarrow 1$ **to** N **do**
5 $y \leftarrow$warp$(t_i \odot x_1, f)$; ▷ Tracker prediction; $\odot$: Hadamard product
6 $t_i \leftarrow y$ if $\phi(y) = 1$; ▷ ϕ : connected region number
7 $M^a \leftarrow M^a \cup \{y\}$ and $M^r \leftarrow M^r \cup \{t_i\}$ if $\phi(y) = 2$;
8 **end**
9 $T \leftarrow T \setminus M^r$; $T \leftarrow T \cup M^a$; $M^a \leftarrow \emptyset$; $M^r \leftarrow \emptyset$;
10 $M \leftarrow$ Hungarian$(T = \{t_i\}_{i=1}^{N'}, B = \{b_j\}_{j=1}^{M})$; ▷ M : match pair indices
11 **for** $i \leftarrow 1$ **to** N', $j \leftarrow 1$ **to** M **do**
12 $t_i \leftarrow b_j$ if (i, j) in M ; ▷ Update
13 $M^a \leftarrow M^a \cup \{b_j\}$ and $M^r \leftarrow M^r \cup \{t_i\}$ if (i, j) not in M;
14 **end**
15 $T \leftarrow T \setminus M^r$; $T \leftarrow T \cup M^a$;

4. Experiments

4.1. Data

The neural cell images used in this work come from a series of time-lapse microscopy videos whose frame size is 512×640. For the training of BSNet, we sample and manually label 386 training images and 129 validation images. The seed heatmap is generated using 2D Gaussian with a radius of 10 centered at the centroid of the blob instances. We use 8 sequences of images with a total number of 8797 neural cell frames to train the UnFlowNet. The tracking performance is evaluated on 9 sequences of images with 1011 frames in total. We use flip, rotation, contrast and brightness distortion for data augmentation.

4.2. Training Details

The BSNet is trained for 100 epochs with a batch size of 16 images, while the UnFlowNet is trained for 60 epochs with a batch size of 12 images. In our experiment we train the BSNet and UnFlowNet independently. Note that we also experiment with sharing the encoder of these two networks, but only observe dramatically decreased performance for the UnFlowNet. Both networks are implemented with PyTorch [29] and run on 4 Nvidia K80 GPUs. We use Adam [22] with an initial learning rate of 0.001 for network optimization. The weights of conv1-conv4 of the encoder (Fig. 2) are initialized from a pretrained ResNet50 [15] on ImageNet datasets. Other parts of the networks are initialized with random weights sampled from a standard Gaussian distribution.

4.3. Evaluation Metrics

The multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA) [4] are employed in this work to evaluate the tracking performance. MOTP reflects the ability of the tracker to estimate the precise object segmentations:

$$\text{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}, \tag{4}$$

where c_t is the number of matches found at time t. For each of these matches, d_t^i is the mask IoU between object o_i and its corresponding hypothesis. MOTA accounts for all object configuration errors made by the tracker, false positives, misses and mismatches over all frames:

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \tag{5}$$

where m_t, fp_t and mme_t are the number of misses (or false negatives, FN), false positives (FP) and mismatches, respectively. g_t is the number of objects present at time t.

Note that in Table 1, FP is the number of false detection, FN is the number of missed detection, TP is number of correct detection. We use N_GT as the total number of cells in the 9 sequential testing images. ID_{sw} is the number of tracker ID switches. We follow the same strategies in [4] to calculate these variables. Briefly speaking, if no blob mask in tracker sets T_s matches with a groundtruth mask at time s (i.e., mask IoU< 0.3), we count the case as a FN. If a blob mask in the tracker sets T_s has no matched groundtruth mask, we count the case as a FP. Otherwise, if a blob mask in T_s is matched with a grountruth mask, we count the case as a TP and we create a correspondence between them. If a new correspondence is made at time $s + 1$ which contradicts the old correspondence, we count this case as a ID_{sw} and we update the correspondence.

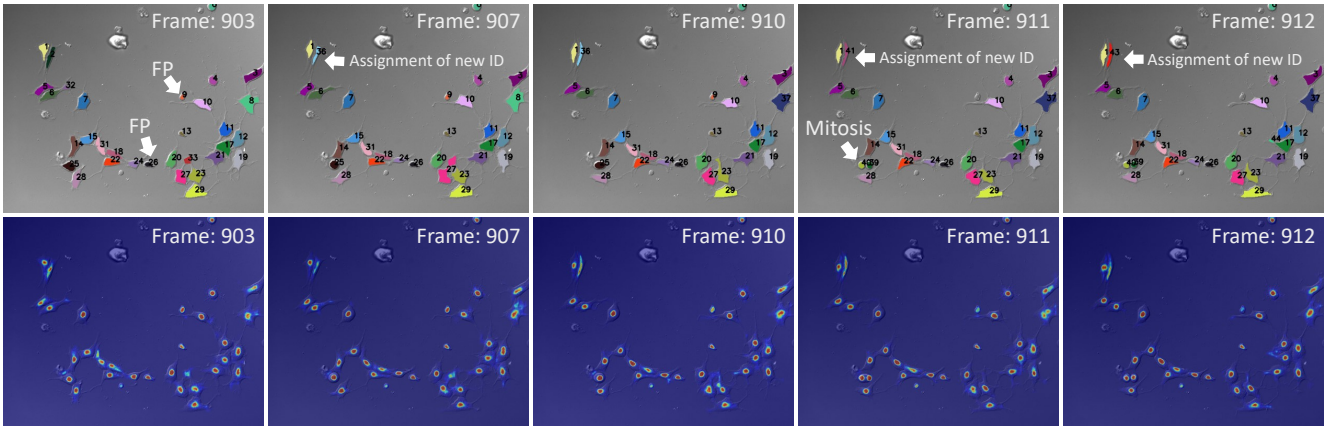

Figure 3. Qualitative examples of tracking results. The top row shows the images that are projected with their blob instance segmentation masks. The bottom row shows the related seed heatmaps.

Method	MOTA ↑	MOTP ↑	N_GT	TP ↑	FP ↓	FN ↓	ID_{sw} ↓
Watershed+Kalman	-0.0732	0.5018	11879	4684	5259	7195	295
BNet+Kalman (Ours)	0.8235	0.9754	11879	10646	768	1233	96
BNet+UnFlowNet (Ours)	0.8737	0.9729	11879	11421	960	458	82
BSNet+UnFlowNet (Ours)	0.8746	0.9744	11879	11685	1215	194	81

Table 1. Tracking evaluation results. N_GT is the total number of cells in the 9 sequences of testing images. ID_{sw} represents the number of tracker ID switches. TP, FP and FN indicate the number of true positive, false positive and false negative, respectively. BNet is the BSNet without seed heatmaps.

4.4. Results

The evaluation results of our tracking methods are shown in Table 1. First, we investigate the effect of seed heatmap. From the ablation study between row3 and row4 of Table 1, we observe that the seed heatmap helps to decrease the false negatives due to the separation of attached cells, while it would also lead to false positives. The reason would be that the seed heatmap can hardly predict the seed for neural cells with extremely irregular morphological shapes, such as those with a long extension (see Fig. 3). However, from TP we find that the seed heatmap is helpful in locating the target cells. As a result, the overall performance (i.e., MOTA) of BSNet+UnFlowNet is better than that of BNet+UnFlowNet. We also compare our UnFlowNet performance with Kalman filter [19]. From the results of row2 and row3 in Table 1, it can be observed that UnFlowNet suppresses the number of false negatives significantly, indicating that UnFlowNet is better at identifying the newly appeared cells from mitosis. Note that, although Kalman filter has a stronger ability to reduce false positives, overall its performance is inferior to UnFlowNet, as shown by the MOTA score. Finally, we compare our BNet with the traditional watershed-based neural cell tracking approach [2]. The results demonstrate the significant advantage of our method. Some qualitative tracking results are illustrated in Fig. 3. As can be seen, there are some background impurities (#9 and #26) which are very similar to cells in appearance and thus are recognized as cells. Besides, for the same cell, its ID could be updated constantly through different frames. This phenomenon is caused by the failure of seed detection. In particular, the inaccurate seed prediction in frame x_1 would lead to the inability of our method to separate touching cells. Consequently, these touching cells are treated as one instance and the additional tracker IDs will be removed from T. Although in the succeeding frame x_2 the touching cells are correctly separated, the newly emerged instances are assigned with new tracker IDs.

5. Conclusion

In this paper, we propose an online neural cell tracking method that exploits both the appearance and motion features of neural cells. The appearance information is captured by BSNet, and the motion feature is extracted using UnFlowNet. Compared to previous methods, our method is able to achieve higher tracking accuracy.

References

[1] S. U. Akram, J. Kannala, L. Eklund, and J. Heikkilä. Joint cell segmentation and tracking using cell proposals. In *ISBI*, pages 920–924. IEEE, 2016.

[2] O. Al-Kofahi, R. J. Radke, S. K. Goderie, Q. Shen, S. Temple, and B. Roysam. Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. *Cell cycle*, 5(3):327–335, 2006.

[3] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *TPAMI*, 33(9):1806–1819, 2011.

[4] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, pages 850–865. Springer, 2016.

[6] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In *ICIP*, pages 3464–3468. IEEE, 2016.

[7] J.-Y. Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *TPAMI*, 33(9):1820–1833, 2011.

[9] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, pages 3029–3037, 2015.

[10] T. Chunming and B. Ewert. Automatic tracking of neural stem cell. *WDIC*, pages 61–66, 2005.

[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015.

[12] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, pages 363–370. Springer, 2003.

[13] N. J. Gordon, D. J. Salmond, and A. F. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F-radar and signal processing*, volume 140, pages 107–113. IET, 1993.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[16] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[17] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, volume 2, page 6, 2017.

[18] J. Y. Jason, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *ECCV*, pages 3–10. Springer, 2016.

[19] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.

[20] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *ICCV*, pages 4696–4704, 2015.

[21] S. Kim, S. Kwak, J. Feyereisl, and B. Han. Online multi-target tracking by large margin structured learning. In *ACCV*, pages 98–111. Springer, 2012.

[22] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2015.

[23] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[24] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 734–750, 2018.

[25] L. Leal-Taixé, C. Canton-Ferrer, and K. Schindler. Learning by tracking: Siamese cnn for robust target association. In *CVPR Workshops*, pages 33–40, 2016.

[26] K. E. Magnusson, J. Jaldén, P. M. Gilbert, and H. M. Blau. Global linking of cell tracks using the viterbi algorithm. *TMI*, 34(4):911–929, 2015.

[27] S. Meister, J. Hur, and S. Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018.

[28] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[30] A. Pinidiyaarachchi and C. Wählby. Seeded watersheds for combined segmentation and tracking of cells. In *ICIAP*, pages 336–343. Springer, 2005.

[31] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, pages 1201–1208. IEEE, 2011.

[32] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha. Unsupervised deep learning for optical flow estimation. In *AAAI*, volume 3, page 7, 2017.

[33] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015.

[34] J. Son, M. Baek, M. Cho, and B. Han. Multi-object tracking with quadruplet convolutional neural networks. In *CVPR*, pages 5620–5629, 2017.

[35] C. Tang and E. Bengtsson. Segmentation and tracking of neural stem cell. In *ICIC*, pages 851–859. Springer, 2005.

[36] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. Torr. End-to-end representation learning for correlation filter based tracking. In *CVPR*, pages 5000–5008. IEEE, 2017.

[37] Y. Wang, Y. Yang, Z. Yang, L. Zhao, and W. Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, pages 4884–4893, 2018.

[38] P. Wu, J. Yi, G. Zhao, Z. Huang, B. Qiu, and D. Gao. Active contour-based cell segmentation during freezing and its application in cryopreservation. *TBME*, 62(1):284–295, Jan 2015.

[39] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *ICCV*, pages 4705–4713, 2015.

[40] J. Yi, P. Wu, D. J. Hoeppner, and D. Metaxas. Fast neural cell detection using light-weight ssd neural network. In *CVPR Workshops*, July 2017.

[41] J. Yi, P. Wu, D. J. Hoeppner, and D. N. Metaxas. Pixel-wise neural cell instance segmentation. In *ISBI*, pages 373–377, April 2018.

[42] J. Yi, P. Wu, Q. Huang, H. Qu, D. J. Hoeppner, and D. N. Metaxas. Context-refined neural cell instance segmentation. In *ISBI*, 2019.

[43] J. Yi, P. Wu, M. Jiang, D. J. Hoeppner, and D. N. Metaxas. Instance segmentation of neural cells. In *ECCV Workshops*, September 2018.

[44] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, pages 1–8. IEEE, 2008.