SEGMENTATION AND TRACKING OF *PSEUDOMONAS AERUGINOSA* FOR CELL DYNAMICS ANALYSIS IN TIME-LAPSE IMAGES

Jianxu Chen[⋆] *Yiqing Cai*[†] *Chen Wei*[‡] *Lin Yang*[⋆] *Mark S. Alber*[♮,⋆] *Danny Z. Chen*[⋆]

[⋆] Dept. of Computer Science and Engineering, University of Notre Dame, USA
[†] Dept. of Biomedical Engineering, Zhejiang University, China
[‡] Dept. of Control Science and Engineering, Zhejiang University, China
[♮] Dept. of Applied and Computational Mathematics and Statistics, University of Notre Dame, USA

ABSTRACT

The swarming behavior of *Pseudomonas aeruginosa* is an important factor to the formation of bacteria biofilm which may cause infections in susceptible individuals. A key issue in studying the mechanism of *P. aeruginosa* swarming is to analyze the cell dynamics in time-lapse images. We propose a complete toolkit for producing quantitative results of cell dynamics analysis in time-lapse images of *P. aeruginosa*. The toolkit includes automatic cell segmentation and tracking, and optional tools for human interaction. Our proposed segmentation and tracking algorithms achieve considerable improvement over one of the best known open-source software for cell segmentation and tracking. A set of graphical interfaces is developed for human annotation, segmentation and tracking results inspection and correction, which are generic and can be combined with most other cell segmentation and tracking algorithms. Our toolkit is convenient for scientific researchers to use, even without knowledge of image processing, to perform cell dynamics analysis efficiently.

Index Terms— Cell Segmentation, Cell Tracking, Pseudomonas Aeruginosa, Graphical User Interface

1. INTRODUCTION

Pseudomonas aeruginosa is a common pathogen causing hospital-acquired infections. Evidences showed that the highly coordinated swarming of *P. aeruginosa* enables cells to colonize tissue surfaces, which may eventually result in infections. However, the swarming mechanism is not fully understood. Thus, segmentation and tracking of *P. aeruginosa* in image sequences acquired from time-lapse experiments is fundamental in various biomedical and clinical studies.

Due to heavy workload and poor reproducibility of manual processing, (semi-)automatic tools, such as [1], have been developed to automate the processing. Many algorithms were proposed [2], but rarely for *P. aeruginosa*. Common approaches may suffer troubles due to the specific characteristics of *P. aeruginosa*. For instance, some cells may move so fast that the same cell may have no overlap in consecutive frames. This makes most model evolution based methods [3] (usually conducting segmentation and tracking simultaneously) not applicable. Moreover, *P. aeruginosa* cells can touch so tightly that the appearance of some head-to-head touching cells is similar to single cells (see Fig. 2(a),(c)). As a result, both falsely merged and falsely split segmentation may occur frequently. Note that classic touching cell separation methods (e.g., watershed) may not be a feasible solution due to the elongated shape of *P. aeruginosa*. In such scenarios, traditional detection association based methods may not work well. In [4, 5], a hierarchical tracking scheme based on an Earth Mover's Distance (EMD) matching model was proposed and shown to be insensitive to segmentation errors.

Human intervention, when necessary, can be applied to ensure or improve accuracy. However, most of the known manual segmentation and tracking tools [1, 6] are unportable and thus hard to integrate with other automatic components. Such tools may be very handy, but running without automated procedures could still be terribly time-consuming.

In our method, we first perform a classic pixel classification based segmentation and a post-processing specifically for *P. aeruginosa*. Then, we propose a new EMD matching model for *P. aeruginosa*, which fundamentally generalizes the EMD model in [4]. Our new EMD model inherits from [4] the robustness to segmentation errors and the capability of handling different cell behaviors. Meanwhile, we introduce a new concept, *edge factors*, in the model. The edge factors serve as a medium for incorporating prior domain knowledge into account or leveraging the power of machine learning, and consequently make our new EMD matching model applicable to most types of cell tracking applications. (The model in [4] only works for certain cell types.) Besides, the *ground distance*, a key component of EMD, is extended to handle general shapes, while that in [4] was primarily designed for curvilinear objects. Furthermore, we develop a set of generic tools for manual inspection and correction of the segmentation and tracking results. It can be seamlessly combined with most of cell segmentation or tracking algorithms and used in different applications. Our objective is to build a general

978-1-4799-2349-6/16/$31.00 ©2016 IEEE

platform so as to allow users to apply specific algorithms for different applications. The entire toolkit is available at `www3.nd.edu/~jchen16/project/cMark.html`.

2. METHODOLOGY

In this section, we present a complete pipeline, which takes the raw time-lapse images of *P. aeruginosa* as input and finally produces quantitative results for studying cell dynamics (see Fig. 1). We will discuss first our automatic steps, i.e., cell segmentation and tracking, and then the manual steps, i.e., human annotation, inspection, and correction.

2.1. Atuomatic Steps

Cell Segmentation. We adopt a cell segmentation algorithm based on pixel classification [1] and design specific post-processing algorithms. For each pixel, 62 features are computed for its classification by the random forest method. The features include those used in [1], the intensity after top-hat transformation, the optimal response and scale of a multi-scale Frangi filter [7], and the local orderless measure. The Frangi filter is designed for tubular structure enhancement. We apply Frangi filter at the scales 1, 3, 5, 7, and 9. For each pixel, the maximum response of all scales, and the scale achieving maximality are selected as features. Frangi filter also estimates the local orientation of the tubular structure. Thus, within a 5×5 neighbor of each pixel, we evaluate how the estimated orientations align (i.e., the magnitude of the average orientation), called the local orderless measure. For a pixel away from tubular structures, the estimated orientation of the neighboring pixels could be very orderless, and the magnitude of the average orientation tends to vanish.

Based on the pixel classification results, a post-processing is conducted, as follows. First, a size threshold is applied to remove small objects, followed by hole filling to make each detected region simply-connected. Next, we try to break the potential false connections between proximal cells. The objective is to make the segmentation as accurate as possible. Even some cuts are made incorrectly or are missing, our tracking algorithm is capable of handling such errors. Since *P. aeruginosa* is of a rod-like shape with uniform thickness, a false connection always appears as a tenuous "bridge" between two thick rod-like parts (see Fig. 2(d)). Thus, we extract the skeleton of each connected region in the image. For every point on a region skeleton, we compute its shortest distance to the region boundary. Let the smallest shortest distance and average shortest distance of all skeleton points of a region be m and v, respectively. If $m/v < 0.5$, a morphological *open* operation is performed on this connected region with a disk of radius $m+1$ as the structuring element. Finally, the resulting binary image is taken as the segmentation result.

Cell Tracking. We propose a generalized EMD matching model to build correspondence between the two sets of cells in every two consecutive frames, $\mathcal{P} = \{p_1, \ldots, p_n\}$ and $\mathcal{Q} = \{q_1, \ldots, q_m\}$. This EMD matching model is formulated as the following optimization problem. $\hat{\mathcal{P}} = \mathcal{P} \cup p_0$ and $\hat{\mathcal{Q}} = \mathcal{Q} \cup q_o$, where p_0 and q_0 serve as virtual cells in the model to accommodate excess flows due to leaving/entering cells, segmentation errors, or significant size changes.

$$EMD(\hat{\mathcal{P}}, \hat{\mathcal{Q}}) = \min_{f_{ij}} \sum_{0\le i\le n} \sum_{0\le j\le m} D_{ij} f_{ij} \tag{1}$$

subject to

$$\sum_{0\le j\le m} f_{ij} \cdot wp_{ij} = 1\,, \forall\, 1 \le i \le n \tag{2a}$$

$$\sum_{0\le i\le n} f_{ij} \cdot wq_{ij} = 1\,, \forall\, 1 \le j \le m \tag{2b}$$

$$f_{ij} \ge 0\,, \forall\, 0 \le i \le n\,,\ 0 \le j \le m \tag{2c}$$

Essentially, the above EMD matching model has the same spirit as a transportation problem. Let each $p_i \in \mathcal{P}$ be a supplier and $q_j \in \mathcal{Q}$ be a consumer, both with a unit capacity; p_0 and q_0 have unlimited capacity; f_{ij} is the amount of flow from p_i to q_j. D_{ij}, called the *ground distance*, measures the cost of one unit of flow from p_i to q_j. wp_{ij} and wq_{ij}, called the *edge factors*, are real values between 0 and 1. One flow unit from p_i to q_j will consume wp_{ij} of p_i's capacity and satisfy wq_{ij} of q_j's demand. Thus, constraints (2a) and (2b) require the capacity of each p_i ($i > 0$) to be fully consumed and the demand of each q_j ($j > 0$) to be totally satisfied. By solving the above optimization problem, the cell correspondence can be interpreted from the resulting optimal flows: p_i matches with q_j if $f_{ij} > \delta$. Here, δ is set empirically as 0.55.

The ground distance between actual cells (i.e., both $i, j > 0$) measures the cost of p_i matching with q_j, and is formulated as a circular order-preserving assignment problem [8]. Cells p_i and q_j are represented by boundary contours, i.e., two circular lists of discrete points. First, we find the optimal circular order-preserving matching between the two lists, allowing 10% of the points unmatched. Then, we compute the distance between each matched point pair. Finally, we let D_{ij} be the average distance between all matched point pairs.

The ground distance between an actual cell p_i (resp., q_j) and a virtual cell q_0 (resp., p_0) measures the cost of p_i (resp., q_j) matching with nothing, maybe due to cell leaving (resp., entering) the imaging window or segmentation errors. In this case, D_{ij} is defined in three different scenarios. If an actual cell is close to image borders (defined by a distance threshold), then the ground distance is defined as the maximum of the distances from all pixels of that cell to the nearest image border. If the actual cell is far away from image borders, we need to check the "difficulty" of the cell moving in/out of the imaging window. For *P. aeruginosa*, it is less likely for a cell surrounded by many cells to move in/out quickly. Thus, we define *feasible paths* from a cell to image borders. Let X be the centroid of the cell. A line passing through X whose

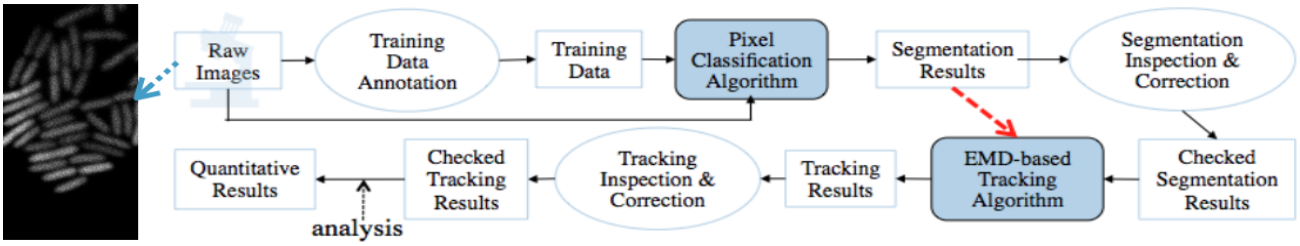

Fig. 1. Our complete pipeline for analyzing time-lapse images of *P. aeruginosa*. A sample image is shown on the left. The red arrow shows a short cut in the pipeline. Because our tracking algorithm is not sensitive to segmentation errors and can automatically resolve certain types of errors, manual segmentation correction could be skipped.

orientation is the same as the estimated cell orientation intersects the image borders at two places, say Y_1 and Y_2. If the line segment XY_i has no intersection with any cells other than that containing X, we call XY_i a *feasible path*. If the actual cell has at least one feasible path, then the ground distance is defined as the length of the (shorter) feasible path, plus half the cell length. Otherwise, the ground distance is set as a large value (e.g., 500) to penalize its (dis)appearance.

An edge between p_i and q_j is associated with two edge factors, wp_{ij} (the consuming ratio of p_i's capacity) and wq_{ij} (the feeding ratio of q_j's demand). If $i = 0$ or $j = 0$, then $wp_{ij} = wq_{ij} = 0$. Otherwise, the edge factors are defined as follows. Let S_{p_i} and S_{q_j} be the sizes (i.e., the number of pixels of a cell) of p_i and q_j, respectively, and $w = S_{p_i}/S_{q_j}$. $wp_{ij} = 1$, $wq_{ij} = w$, if $0 < w < \gamma$; $wp_{ij} = 1/w$, $wq_{ij} = 1$, if $w > 1/\gamma$; otherwise, $wq_{ij} = wp_{ij} = 1$. Here, γ is a threshold value between 0 and 1, indicating the maximum allowable size change of cells in consecutive image frames.

In practice, we conduct a pre-matching to reduce computation time. (p_i, q_j) $(i, j > 0)$ is a *perfect match* if $S_{p_i}/S_{q_j} \in [1-\epsilon, 1+\epsilon]$ and $D_{ij} < \delta$, where ϵ and δ are parameters determined empirically. Smaller ϵ and δ result in a more conservative pre-matching. Ruling out all perfect matches, our matching model is applied to the remaining cells. This process is repeated on every two consecutive frames sequentially.

Our generalized EMD matching model is able to handle many-to-one and one-to-many matchings with proper edge factors. Such flexibility enables our algorithm to handle cell division/fusion (although no fusion is in our datasets) and tolerate falsely merged (FM) or falsely split (FS) segmentation. To recover from FM or FS errors automatically, we apply two post-processing steps every time after matching two consecutive frames. The idea is sketched below.

The first step is FM/FS checking. For *P. aeruginosa*, FM errors occur only when cells with similar orientations touch head-to-head. For example, p_{i_1} and p_{i_2} match to q_j $(i_1, i_2, j > 0)$. First, we determine that either this is an association error or q_j is falsely merged. We check the difference among the orientation of the line segment connecting the centroids of p_{i_1} and p_{i_2}, the orientation of p_{i_1}, and the orientation of p_{i_2}. If they differ less than θ, a pre-determined

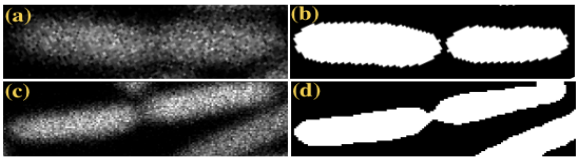

Fig. 2. Examples of segmentation errors. (a)-(b) A raw image and segmentation of one cell with low intensity in the middle, a false split error. (c)-(d) A raw image and segmentation of two cells touching head to head, a false merge error.

threshold, then q_j is cut at the thinnest position; otherwise, q_j is associated with only p_{i_1} or p_{i_2} with the smaller ground distance. On the other hand, one-to-many correspondence may find falsely split segmentations. For instance, p_i matches to q_{j_1} and q_{j_2}. We perform the orientation checking on q_{j_1} and q_{j_2} analogously to the many-to-one case. If the orientation difference is smaller than θ, then q_{i_1} and q_{i_2} are merged by a morphological *close* operation; otherwise, p_i is associated with only q_{j_1} or q_{j_2} with the smaller ground distance.

Certain correspondence errors may still not be addressed by the FM/FS checking. For instance, p_1 and p_2 match to q_1, which is an association error, while p_1 and p_2 should match to q_1 and q_2, respectively. After the FM checking, p_1 is correctly associated with q_1, but p_2 is still unmatched. For this reason, the second post-processing step is to perform the EMD matching again on all unmatched cells between the two consecutive frames to build the missing links.

2.2. Manual Steps

We provide neat graphical user interfaces (GUIs) to assist human interaction/intervention. The **Training Data Annotation GUI** allows users to manually label training data for any pixel-based classification problem. The process is as simple as labelling different classes by drawing lines in different colors. The **Segmentation Inspection and Correction GUI** is to manually check and, if necessary, correct the segmentation results. The segmented regions are color-coded to indicate connectivity and overlaid on the grayscale image for accurate examination. Supported operations include adding new cells

(drawing), trimming cells (erasing), extending cells (drawing), and deleting whole cells (1-click inside a cell region). The **Tracking Inspection and Correction GUI** is designed for inspecting and correcting tracking results. For easy inspection, the correspondences are color-coded. That is, each trajectory is assigned a unique color so that the same cell in different frames is labeled by the same color. Corrections can be made by adding or deleting correspondences.

3. EXPERIMENTS

We conducted experiments on eight different movies of *P. aeruginosa*. Cells were tagged with green fluorescent protein. Our method is compared with ilastik [1], one of the best known open-source software for cell segmentation and tracking, which uses a pixel-classification based segmentation algorithm (with a nice annotation tool) and adopts the algorithm in [9] to build cell correspondence over time.

To evaluate the performance, we employ the TRA measure used in the ISBI cell tracking challenge [2]. One objective of our work is to develop (semi-)automatic algorithms to segment cells in space and track cells in time, so as to reduce or eliminate human effort to obtain all trajectories for cell dynamics analysis. Precisely, TRA computes the minimum human effort to transform the automatic segmentation and tracking results to the ground truth (GT), taking both segmentation errors and tracking errors into account. Let R be a ground truth cell and S be a segmented cell. R corresponds to S if $\|R \cap S\|/\|R\| > 0.5$. S is a true positive (TP), falsely merged (FM), or false positive (FP) segmentation, if S has a unique, multiple, or no corresponding GT cell(s), respectively. A false negative (FN) segmentation occurs if there exists a GT cell with no corresponding segmentation. TRA is a weighted sum of the least numbers of allowable operations, divided by the total number of GT cells in the whole image sequence. A special case is false split (FS) errors (see Fig. 2(b)). Since in the TRA measure, each GT cell has at most one corresponding detection, an FS error may be counted as "one FP" or "two FP + one FN", depending on the sizes of the falsely split cells. The results are summarized in Table 1 (a smaller TRA means less manual operations needed). It is evident that our approach achieves much higher accuracy than [1] and is able to reduce manual effort substantially.

4. CONCLUSIONS

In this paper, we present a complete pipeline for segmenting, tracking, and analyzing *P. aeruginosa* cells in time-lapse images. Evaluated by eight different datasets, our proposed automatic algorithms achieve much higher accuracy than one of the best known open-source software for cell segmentation and tracking. When necessary, human intervention can be easily conducted. A set of handy and portable GUIs (freely available) is developed for results inspection and correction.

Dataset	# of GT Cells	# of Frames	Ours	[1]
1	876	43	0.035	0.862
2	786	43	0.028	0.789
3	905	43	0.153	0.396
4	1082	43	0.030	1.566
5	846	43	0.015	0.648
6	2331	67	0.145	0.169
7	2438	85	0.031	0.442
8	3238	85	0.098	0.984

Table 1. Experimental results. The TRA measure of the results of our method and [1] are shown in the last two columns.

5. ACKNOWLEDGEMENT

This work was partly supported by NIH grant 1R01GM095959 and NSF Grant CCF-1217906. We would like to thank Morgen Anyan and Prof. Joshua Shrout for providing microscopic images.

6. REFERENCES

[1] C. Sommer, C. Straehle, U. Koethe, and F. Hamprecht, "ilastik: Interactive learning and segmentation toolkit," in *ISBI*, 2011, pp. 230–233.

[2] M. Maška et al., "A benchmark for comparison of cell tracking algorithms," *Bioinformatics*, vol. 30, no. 11, pp. 1609–1617, 2014.

[3] M. Maška, A. M. Barrutia, and C. O. de Solórzano, "Fast tracking of fluorescent cells based on the chan-vese model," in *ISBI*, 2012, pp. 1316–1319.

[4] J. Chen, C. Harvey, M. Alber, and D. Z. Chen, "A matching model based on earth mover's distance for tracking myxococcus xanthus," in *MICCAI*, 2014, pp. 113–120.

[5] J. Chen, S. Mahserejian, M. Alber, and D. Z. Chen, "A hybrid approach for segmentation and tracking of myxococcus xanthus swarms," in *MICCAI*, 2015, pp. 284–291.

[6] J. Schindelin et al., "Fiji: An open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7, pp. 676–682, 2012.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *MICCAI*, 1998, pp. 130–137.

[8] C. Scott and R. Nowak, "Robust contour matching via the order-preserving assignment problem," *IEEE Trans. on Image Processing*, vol. 15, no. 7, pp. 1831–1838, 2006.

[9] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. Hamprecht, "Conservation tracking," in *ICCV*, 2013, pp. 2928–2935.