

TRACKING OF CELLS IN A SEQUENCE OF IMAGES USING A LOW-DIMENSION IMAGE REPRESENTATION

Maël Primet¹, Alice Demarez^{1,2}, François Taddei², Ariel B. Lindner², Lionel Moisan¹

¹Paris Descartes University, MAP5, CNRS UMR 8145, Paris, France

²INSERM U571 and Faculty of Medicine, Paris Descartes University, F-75015 France

ABSTRACT

We propose a new image analysis method to segment and track cells in a growing colony. By using an intermediate low-dimension image representation yielded by a reliable over-segmentation process, we combine the advantages of two-steps methods (possibility to check intermediate results) and the power of simultaneous segmentation and tracking algorithms, which are able to use temporal redundancy to resolve segmentation ambiguities. We improve and measure the tracking performances with a notion of decision risk derived from cell motion priors. Our algorithm permits to extract the complete lineage of a growing colony during up to seven generations without requiring user interaction.

Index Terms— Image sequence analysis, Machine vision, Image segmentation, Tracking, Biological cells

1. INTRODUCTION

Image sequences are commonly used by biologists to study living cell dynamics (see for instance the study of cell ageing in [1]). In order to produce quantitative and statistically relevant results, large amounts of data are required (say, several sequences, each containing several hundredth of images), and automatic image analysis algorithms become necessary. One frequent aim is to extract from the image sequence the complete description of cell positions, shape and motion across time, leading, in the case of dividing cells, to a space-time lineage. These segmentation and dividing/tracking issues have to be solved in the most reliable manner, since human post-processing is the limiting factor of the rate of processed data.

Many cell-tracking algorithms (see [2, 3] for recent examples) perform in sequential steps: they generally start by completely segmenting the cells in each frame, then try to track the cells from one image to the next. Analyzing image sequences with such a two-steps process (segmentation, then tracking) has two main advantages: first, it dramatically reduces the combinatorial complexity of tracking, since it works on the object space (the segmented objects) instead of the pixel space. Second, it splits the overall problem in two, and thus produces intermediate results than can be checked and used to improve each step separately. However, in some situations, the segmentation step cannot be achieved in a reliable way on individual images (even the human eye has some difficulties), whereas segmentation ambiguities can be solved by considering the whole sequence. This can be a severe issue in applications where the whole lineage has to be computed exactly, requiring time-consuming human interaction to solve segmentation mistakes. This calls for simultaneous segmentation and tracking algorithms, able to solve segmentation ambiguities using the temporal redundancy of the data.

Some recent approaches [4, 5] build a model of the cell motion in order to improve the segmentation performances. Such one-step

approaches are efficient to resolve segmentation ambiguities when they are localized, or when an accurate motion model can be built. In the case of a bacterial colony, cells are constantly in contact, steadily grow and divide, and push each other, which results in unpredictable rotations and motion. This makes one-step approaches difficult to use, because they do not yield any intermediate representation between image pixels and the final objects (the cells), so that in general there is no easy way to understand what is wrong (and which parts of the algorithm have to be modified) when mistakes are present in the final lineage.

In this paper, we propose a new algorithm that tries to combine the advantages of both approaches by using a two-steps process (so that intermediate results can be checked) while keeping the idea of simultaneous segmentation and tracking. The first step is a reliable and efficient over-segmentation process (described in Section 2) that produces an intermediate low-dimension image representation, a collection of small shapes called *blobs*. The second step is a segmentation-and-tracking iterative algorithm (Section 3), in which the segmentation is performed at the blob level (a cell is a union of connected blobs), and the tracking decisions are ordered with a notion of risk we introduce, which permits to improve the robustness of the process by a factor typically equal to 10. Both steps use no intensity-based thresholds or parameters, but geometric cell properties (minimal and maximal width, minimal area, etc.) and simple parametric cell motion priors. We show experiments made on real data and comment the results obtained.

2. OVER-SEGMENTATION

Since we would like to analyze the image sequence with a high degree of reliability, we have to be very careful in the first step of data processing, since an error at this step will inevitably make the whole algorithm fail. Hence, rather than trying to perform a complete segmentation of each image, we compute an “over-segmentation”, that is, a partition of the image domain into a background domain and small regions called *blobs*, with the properties that any cell of the image is a union of connected blobs, and that any blob belong to exactly one cell. Such an over-segmentation seems to achieve a good compromise, because it manages to simplify the images into a small number of “objects” (the blobs) without having to solve ambiguous decisions.

2.1. Image renormalization

A natural idea to define such blobs is to grow seeds obtained after some gray-level thresholding (the darkest regions of the image are the inside of the cells). Since image illumination may slowly vary inside one image, or may change across time or between ex-

periments, we first apply a gray-level renormalization based only on geometrical and physical assumptions:

- (A1) **There is a minimum width m of a cell**
- (A2) **There is a maximum width M of a cell**
- (A3) **The illumination artifacts are slowly-varying.**

The main idea is to sample the gray level at various places near the cells in the picture, and estimate the local gray-level thresholds that can be used to define seeds (low threshold) and extra-cell space (high threshold).

We first use a Fast Level-Set Transform [6] to obtain what we call *seeds*, that is, connected regions defined by the image lower level-sets with area less than $\pi m^2/4$ (A1) (discarding the very small lower-level sets as noise). We then compute the maximum of u (the original image) on each seed, and extrapolate these values on the whole image domain by using a Gaussian convolution from known values. This process yields a smooth (A3) “low threshold image” v_- .

To compute the “high threshold image” in a similar way, we compute for each point x of each seed, the gray value

$$\rho(x) = \min_{y;x \in B(y,M)} \max_{z \in B(y,M)} u(z),$$

where $B(y, M)$ denotes the disc of center y and radius M . From (A2), we deduce that if $x \in B(y, M)$, then the disc $B(y, M)$ necessarily contains a pixel outside the cell, so that the max over z is an upper bound for the optimal high threshold, that we can minimize with respect to y . We then compute the maximum of $\rho(x)$ on each seed, and extrapolate these values on the whole image domain in the same way as before, yielding a “high threshold image” v_+ .

We finally normalize the original image by a point-wise affine transform $w(x) = (u(x) - v_-(x))/(v_+(x) - v_-(x))$.

2.2. Non-uniform dilation

Now that the seeds have been found and the image renormalized, we grow the seeds into *blobs* by using a concurrent dilation (*ie.* blobs should not penetrate each other) that bears some resemblance with the watershed transform [7]. The dilation is non-uniform as its speed depends on the local gray level (the blobs grow faster in dark areas, and slower in the bright regions that separate two cells, so they cannot cross the cells borders).

Let us precise a little bit the dilation process. We assume that we are given N disjoint sets (the seeds) s_1, \dots, s_N , and a viscosity function $v : \Omega \rightarrow [1, +\infty[$, where Ω is the image domain. This viscosity function will determine the speed of the dilation at any point in space, the higher the value, the slower the dilation. We define paths on Ω as C^1 functions from $[0, 1]$ to Ω , and the length of a path as

$$\delta(\gamma) = \int_0^1 v(\gamma(t)) \|\gamma'(t)\| dt$$

and the distance $\delta(a, b)$ between two points of Ω as the lower bound of the length of a path connecting the two points, and the distance $\delta(a, X)$ between a point of Ω and a set $X \subseteq \Omega$ as the lower bound of the distance between a and a point of X . We then define the non-uniform dilation of the sets s_1, \dots, s_N as the sets (the blobs) b_1, \dots, b_N where

$$b_i = \{ x \in \Omega \mid \forall j \neq i, \delta(x, s_i) < \delta(x, s_j) \}$$

We can directly translate this definition in the discrete domain, and compute the resulting blobs efficiently using an operation akin to a

dilation. In experiments, we limited the dilation process to a fixed maximum length τ and chose an arbitrary (but reasonable) viscosity function (recall that since the images are normalized, the viscosity function is not image-dependent but defined on an absolute scale).

2.3. Blobs and connection graph simplification

After we get the initial blobs, which represent an over-segmentation of the cells, we can build some kind of under-segmentation to limit our possible choices.

It is obvious that each blob that is not already a complete cell must be connected to some neighboring blobs. Following this idea, we define a connection graph, where the blobs are the vertices, and edges link neighboring blobs (that is, blobs that share a long enough boundary). Possible cells are then “linear sub-graphs” (homeomorph to a segment) of this connection graph. However, not all edges in the connection graph are meaningful, and many of them can be removed using some simple conservative criteria that ensure the correctness of the new over-segmentation. The assumptions we use are (A2)¹ and

- (A4) **There is a minimum area A of a cell.**

By iterating the processes of removing edges in the connection graph using assumption (A2) and merging the blobs using assumption (A4) (a blob that is too small to be a cell on its own and that only has one neighbor must be merged with its neighbor), we achieve a significant simplification of the over-segmentation, as shown on Fig. 1.

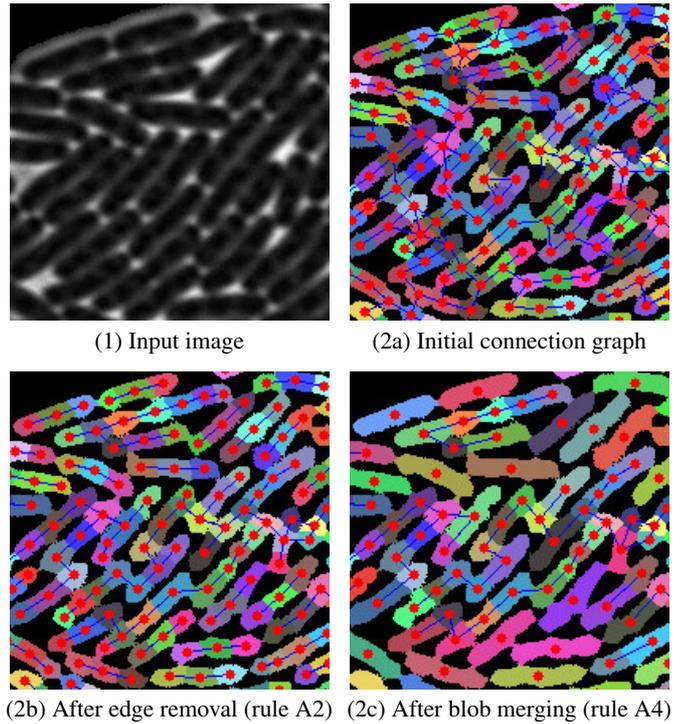


Fig. 1. A typical cell image (1), the initial connection graph of blobs (2a), and the two steps of the graph simplification process (2b and 2c), iterated until convergence. Note that the isolated blobs of the graph (2c) necessarily are cells.

¹Note that in order to accommodate for “bent” cells in (A2), the width is defined in term of the minimal-width enclosing annulus.

3. CELL SEGMENTATION AND TRACKING

Now that we have an over-segmentation and an under-segmentation, we can generate all potential cells that are consistent with these bounds – in other words, all the unions of blobs on “linear sub-graphs” of the connection graph described in the previous section. Some of these potential cells are necessarily true cells, as they are isolated vertices of the connection graph. In general, there are enough true cells after the blob simplification process, but we can always assume that one or two images (say the first and the last) have been manually segmented, and only contain perfectly segmented cells, so we obtain enough information to start the segmentation and tracking algorithm.

3.1. Motion likelihood

Given a cell A in image n and a cell B in image $n + 1$, we need to rate the likelihood of the hypothesis $A \rightarrow B$ (that is, A becomes B). To do this, we extract from each cell C three parameters: its position $x_C \in \mathbb{R}^2$ (the origin being the center of mass of the image), its area $\mathcal{A}_C \in \mathbb{R}_+$, and its orientation $\theta_C \in S^1$. We then model the probability density of transition from A to B by

$$\pi_{A \rightarrow B} = (1 - \pi_{\text{div}}) \cdot \pi_x \left(\frac{x_A - x_B}{|x_A|} \right) \cdot \pi_\theta (|\theta_A - \theta_B|) \cdot \pi_{\mathcal{A}} \left(\frac{\mathcal{A}_A}{\mathcal{A}_B} \right),$$

where the probability that a cell divides (π_{div}) is determined empirically (it depends on the frame rate) and the probability densities π_x , π_θ and $\pi_{\mathcal{A}}$ are designed according to biological knowledge and their parameters are learned from (previously processed) reference sequences (see Fig. 2). Concerning the speed (π_x), we chose to measure the relative motion $(x_A - x_B)/|x_B|$ instead of the absolute motion $x_A - x_B$, as the cell motion results from the cells in the center of the colony pushing the other ones because of their growth, so that we expect the motion amplitude to be roughly proportional to the distance to the center of mass of the colony.

To define a similar probability density for the transition $A \rightarrow B, C$ (A divides into B and C), we note that if the frame rate is not too small, the union of the two new cells (B and C) can be considered as a single cell when the parameters x , \mathcal{A} and θ are measured, so that it is natural to write $\pi_{A \rightarrow B, C} = \pi_{A \rightarrow B \cup C} \cdot \pi_{\text{div}} / (1 - \pi_{\text{div}})$.

3.2. Likelihood versus risk

By taking the product of the likelihood of all its local transitions $A \rightarrow B$ and $A \rightarrow B, C$, we can define the likelihood of a complete lineage (each cell of the lineage being a union of blobs). Ideally, we would like to find the lineage that has the highest likelihood, with the constraints that each blob belongs exactly to one single cell. However, this global optimization problem seems computationally intractable, and in particular affectation algorithms [8] cannot handle such constraints.

If we resolve to find a lineage by taking local tracking decisions, the most natural way to define the successor of a cell is to associate to a cell its best match in the next frame, so a natural idea would be to start with the most likely decision, then the second possible most likely decision taking into account the first one, etc. However, the tracking decisions are order-dependent, as the best match of a cell could also be the best match of another one, so rather than taking first the most likely decision, we propose to take first the least ambiguous one. To measure the ambiguity of a given transition $A \rightarrow B$ (or, similarly, of a cell division $A \rightarrow B, C$), we introduce the notion of

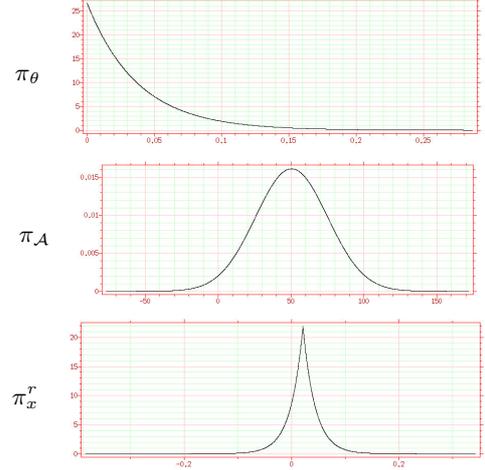


Fig. 2. Learned probability density functions for the cell evolution model: rotational speed (exponential), growth rate (Gaussian) and radial component of the speed (Laplace; the tangential component is similar up to a scale factor).

risk, defined by

$$\rho_{A \rightarrow B} = \max_{X \neq B} \frac{\pi_{A \rightarrow X}}{\pi_{A \rightarrow B}},$$

the maximum being taken over all potential successors X of A (B excepted). Intuitively, the risk is very low when the transition has no credible alternative, and rises when there is a doubt on the successor of the cell. Since most of the cells have a trivial motion, we can hope that they will be processed early and correctly and will rule out some choices concerning other cells for which the initial risk was high (see Fig. 3). Note that this notion of risk is also related to the robustness of the algorithm, since the maximum risk can be understood as the level of degradation that the algorithm can handle before changing its choices.

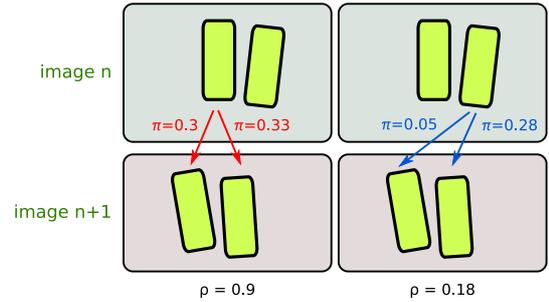


Fig. 3. Example of a dubious (left) and an obvious (right) choice, π is the likelihood of the associated transition, ρ is the risk of the best-match transition. If we take the most likely transition first (left), we will make the wrong affectation, but if we take the most obvious transition first (right), the final result we be correct.

3.3. Tracking segmented cells

To quantify the benefits of using the notion of risk, we implemented three tracking algorithms working on completely segmented

	original	1 frame / 2	1 frame / 4
(1)	$\rho_{\max} = 0.0026$ $\rho_{\text{average}} = 3.2 \cdot 10^{-6}$	0.23 0.001	0.89 0.03
(2)	$\rho_{\max} = 0.0001$ $\rho_{\text{average}} = 2.3 \cdot 10^{-7}$	0.06 0.0003	0.24 0.01
(3)	$\rho_{\max} = 6.9 \cdot 10^{-6}$ $\rho_{\text{average}} = 3.5 \cdot 10^{-8}$	0.004 $7 \cdot 10^{-5}$	0.11 0.005

Table 1. Maximum and average risks encountered by the three algorithms when extracting the lineage from the original movie and various sub-sampled versions. We clearly see that the max and average risks that have been taken are always lower when the "obvious-first" algorithm is used, thus giving an increased confidence in its results.

sequences:

- (1) the *any-first* algorithm, which orders the cells arbitrarily,
- (2) the *likely-first* algorithm, which sorts the cells by decreasing likelihood of their best transition,
- (3) and the *obvious-first* algorithm, which sorts the cells by increasing risk of their best transition.

Each algorithm works in the same way: to each cell in the order given by the algorithm, we associate its best possible match in the next frame, and this process is iterated.

To compare the three algorithms, a sequence of previously segmented cell images was degraded by under-sampling it twice (keeping one frame in two) and four times (one frame in four). On each sequence and for each of the three algorithms, we then computed the average and the maximum risk taken (see Table 1). As can be seen, the more degraded the film, the higher the risk. Algorithm (1) makes some errors when tracking the most degraded film (one frame in four), and the other algorithms always give correct results, although we see that algorithm (3) leads to minimal risks, and thus gives us more confidence in the result of the tracking.

3.4. Tracking over-segmented cells

We apply this risk-based approach to the potential cells defined by the blobs and the connection graph. Among these potential cells are some perfectly segmented cells (isolated vertices of the connection graph), that we label as initial "active" cells. For each active cell, we compute the risk of its associated best transition to any potential cell in the next frame, and then select the transition having the minimal risk among the active cells. The target cell of the selected transition now becomes active, all the potential cells overlapping it are deleted, and we recompute the risks of the transitions for the new set of active cells. This process is then iterated until the complete lineage has been obtained. We also consider the risks of the transitions in the backward direction (knowing the cell, what is its best-match predecessor?) and apply the same process to these transitions.

3.5. Current results

On the films that we processed, we could handle images containing about 120 cells (about 7 generations) before the number of blobs and connections between them was too large for the potential cells to be efficiently generated. Developing discriminatory but conservative shape constraints on the potential cells to simplify the connections between the blobs could break this limit, and is thus an interesting challenge. With the current algorithm we proposed, if we want to

handle films up to 100 frames (9-10 generations), we have to assist the connection graph simplification algorithm by manually deleting some connection edges (or by using supervised heuristics). Such interaction also permits to fix some little segmentation errors, that can propagate and induce tracking errors. On the first frames (about 5 to 6 generations on most films), we have almost no error (see Fig. 4). In the last frame (9-10 generations), the number of cells suddenly increases, so the connection graph becomes very large, and there are a lot of potential cells, which slows down the computations and introduce more errors (when there are many potential cells, there are more cells that "look like good successor").

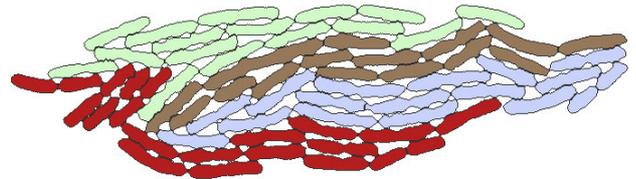


Fig. 4. A part of the result of the tracking algorithm we propose: a completely segmented image, where each cell is colored according to its 3rd generation ancestor. This result was obtained in completely unsupervised way (no human interaction).

We implemented in C (using the MegaWave library) a complete segmentation and tracking suite called CellST that has a user-friendly interface and permits to visualize and correct the results in a straightforward and natural manner. With this software, a complete sequence requires almost no interaction for the 5-6 first generations, and about a couple of hours for 9-10 generations, which seriously improves previously-used algorithms.

4. REFERENCES

- [1] E.J. Stewart, R. Madden, G. Payl, and F. Taddei, "Aging and death in an organism that reproduces by morphologically symmetric division," *PLoS Biol* 3(2): e45, 2005.
- [2] V. Gor, M. Elowitz, T. Bacarian, and E. Mjolsness, "Tracking cell signals in fluorescent images," in *Proc. CVPR'05*, 2005, vol. 03, p. 142.
- [3] C. Tang and E. Bengtsson, "Segmentation and tracking of neural stem cell," in *International Conference on Intelligent Computing (2)*, 2005, pp. 851–859.
- [4] K. Li, E. Miller, L. Weiss, P. Campbell, and T. Kanade, "Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy," in *Proc. of CVPRW'06*, 2006, p. 65.
- [5] K. Althoff, J. Degerman, and T. Gustavsson, "Combined segmentation and tracking of neural stem-cells," in *Scandinavian Conference on Image Analysis*, 2005, pp. 282–291.
- [6] P. Monasse and F. Guichard, "Fast computation of a contrast-invariant image representation," *IEEE Trans. on Image Processing*, pp. 860–872, 2000.
- [7] J. Roerdink and A. Meijster, "The watershed transform: definitions, algorithms and parallelization strategies," *Fundamenta Informatica*, vol. 41, no. 1-2, pp. 187–228, 2000.
- [8] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistic Quarterly*, vol. 2, pp. 83–97, 1955.