

STOCHASTIC NEURAL NETWORKS FOR AUTOMATIC CELL TRACKING IN MICROSCOPY IMAGE SEQUENCES OF BACTERIAL COLONIES

SORENA SARMADI*, JAMES J. WINKLE*, RAZAN N. ALNAHHAS^{†‡}, MATTHEW R. BENNETT^{‡§}, KREŠIMIR JOSIĆ^{*¶}, ANDREAS MANG*, AND ROBERT AZENCOTT*

Abstract. We describe an automated analysis method to quantify the detailed growth dynamics of a population of bacilliform bacteria. We propose an innovative approach to frame-sequence tracking of deformable-cell motion by the automated minimization of a new, specific cost functional. This minimization is implemented by dedicated Boltzmann machines (stochastic recurrent neural networks). Automated detection of cell divisions is handled similarly by successive minimizations of two cost functions, alternating the identification of children pairs and parent identification. We validate this automatic cell tracking algorithm using recordings of simulated cell colonies that closely mimic the growth dynamics of *E. coli* in microfluidic traps. On a batch of 1100 image frames, cell registration accuracies per frame ranged from 94.5% to 100%, with a high average. Our initial tests using experimental image sequences of *E. coli* colonies also yield convincing results, with a registration accuracy ranging from 90% to 100%.

Key words. stochastic neural networks, cell tracking, microscopy image analysis

1. Introduction. Technology advances have led to increasing magnitudes of data generation with increasing levels of precision [25, 32, 82, 90]. Data generation presently far outpaces data analysis, however, and drives the requirement for analyzing such large-scale data sets with automated tools [16, 35, 41, 48, 66, 87, 89]. The main goal of the present work is to develop computational methods for an automated analysis of microscopy image sequences of colonies of *E. coli* growing in a single layer. Such recordings can be obtained from colonies growing in microfluidic devices, and they provide a detailed view of individual cell-growth dynamics as well as population-level, inter-cellular mechanical and chemical interactions [5, 6, 19, 28, 29, 65].

However, to understand both variability and lineage-based correlations in cellular response to environmental factors and signals from other cells requires the tracking of large numbers of individual cells across many generations. This can be challenging, as large cell numbers tightly packed in microfluidic devices can compromise spatial resolution, and toxicity effects can place limits on the temporal resolution of the recordings [33, 40]. One approach to better understand and control the behavior of these bacterial colonies is to develop computational methods that capture the dynamics of gene networks within single cells [5, 19, 46, 94]. For these methods to have a practical impact, one ultimately has to fit the models to the data, which allows us to infer hidden parameters (i.e., characteristics of the behavior of cells that cannot be measured directly). Image analysis and pattern recognition techniques for biological imaging data [26, 43, 66], like the methods developed in the present work, can be used to track lineages and thus automatically infer how gene expression varies over time. These methods can serve as an indispensable tool to extract information to fit and validate both coarse and detailed models of bacterial population, thus allowing us to infer model parameters from recordings.

Here we describe an algorithm that provides *quantitative* information about the population dynamics, including the life cycle and lineage of cells within a population from recordings of cells growing in a mono-layer. A typical sequence of frames of cells growing in a microfluidic trap is shown in Fig. 1. We describe the design and validation of algorithms for tracking individual cells in sequences of such images [5, 46, 55]. After segmentation of individual image frames to identify each cell, tracking individual cells from frame to frame is a combinatorial problem. To solve this problem we take into account the unknown cell growth, cell motion, and cell divisions that occur between frames. Segmentation and tracking are complicated by imaging noise and artifacts, overlap of bacteria, similarity of important cell characteristics across the population (shape; length; and diameter), tight packing of bacteria, and large interframe durations resulting in significant cell motion, and up to a 30% increase in individual cell volume.

1.1. Related Work. The present work focusses on tracking *E. coli* in time series of images. A comparison of different cell-tracking algorithms can be found in [35, 89]. Tracking and object recognition in time series of images is a challenging task that arises in numerous applications in computer vision [63, 97]. In image processing, motion tracking is often referred to as “*image registration*” [35, 57, 51, 60, 62, 61] or “*optical flow*” [23, 39, 47, 50, 95].

Our tests have shown that off-the-shelf image intensity driven techniques fail to provide a robust bacteria cell

*Department of Mathematics, University of Houston, Houston, US

†Department of Biomedical Engineering, Boston University, Boston, US

‡Department of Biosciences, Rice University, Houston, US

§Department of Bioengineering, Rice University, Houston, US

¶Department of Biology and Biochemistry, University of Houston, Houston, US

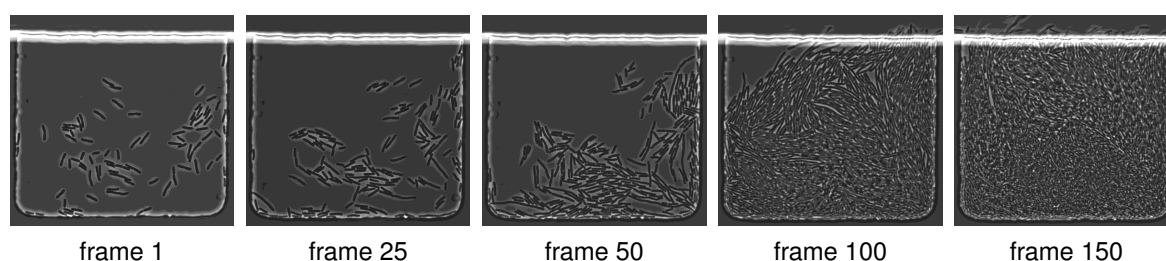


FIG. 1. **Typical microscopy image sequence.** We show five frames out of a total of 150 frames of an image sequence showing the dynamics of *E. coli* in a microfluidic device [5].

registration in the tightly packed colonies of rod-shaped *E. coli* bacteria considered here. Moreover, we are not interested in tracking individual pixels but rather cells (i.e., rod-shaped, deformable shapes), while recognizing events of cell division and recording cell lineage.

One approach proposed in prior work to simplify the tracking task is to make the experimental setup more rigid by confining individual cell lineages to small tubes; the associated microfluidic device is called a “mother machine” [18, 42, 58, 71, 79, 85]. The microfluidic devices we consider here yield more complicated data as cells are allowed to move and multiply freely in two dimensions (constrained to a mono-layer). We refer to Fig. 1 for a typical sequence of experimental images considered in the present work.

Turning to methods that work on more complex biological cell imaging data, we can distinguish different classes of tracking methods. “Model-based evolution methods” operate on the image intensities directly. They rely on particle filters [8, 70, 84] or active contour models [7, 37, 45, 52, 93, 96]. These methods work well if the cells are not tightly packed. However, they may lead to erroneous results if the cells are close together, the inter-cellular boundaries are blurry, or the cells move significantly. Our work belongs to another class—the so called “detection-and-association methods” [17, 22, 20, 44, 74, 78, 88, 92, 98], which first detect cells in each frame and then solve the tracking problem/association task across successive frames. Doing so necessitates the segmentation of cells within individual frames. We refer to [91] for an overview of cell segmentation approaches. Deep learning strategies have been widely used for this task [4, 34, 59, 67, 72, 78, 77, 86, 87, 98]. We consider a framework based on convolutional neural networks (CNNs). Others have also used CNNs for cell segmentation [3, 59, 69, 76, 77]. We omit a discussion of our segmentation approach, as we do not view it as our main contribution (see Sec. 1.2). To solve the tracking problem after the cell detection, many of the methods cited above use hand-crafted association scores based on the proximity of the cells and shape similarity measures [44, 22, 88, 98]. We follow this approach here. We note that we not only consider local association scores between cells but also include measures for the integrity of a cell’s neighborhood (i.e., “context information”).

Our method is tailored for tracking cells in tightly packed colonies of rod-shaped *E. coli* bacteria. This problem has been considered previously [17, 74, 87, 92]. However, we are not aware of any large-scale datasets that provide ground truth tracking data for these types of recordings, but note that there are community efforts for providing a framework for testing cell tracking algorithms [64, 89].* Works that consider these data are for example [8, 59, 69, 73, 72, 98]. The cells in this dataset have significantly different characteristics compared to those considered in the present work. As we describe below, our approach is based on distinct characteristics of the bacteria cells and, consequently, does not directly apply to these data. Therefore, we have developed our own validation and calibration framework (see Sec. 2).

[27, 54, 53, 56] considered graph-based matching strategies for global association. Similar to the methods described above, they used association scores for tracking. Individual cells are represented as nodes, and neighbors are connected through edges. This is similar to our approach in that we construct local neighborhood relations based on a (modified) Delaunay triangulation. By using a graph-like structure, cell divisions can be identified by detecting changes in the topology of the graph [54, 53]. We tested a similar strategy, but came to the conclusion that we cannot reliably construct neighborhood networks between frames for which topology changes only occur due to cell division; the main issue we observed is that the significant motion of cells between frames can introduce additional topology changes in our neighborhood structure. Consequently, we decided to relax these assumptions.

Some recent works jointly solve the tracking and segmentation problem [8, 36, 73, 74, 72, 98]. Contrary to observations we have made in our data, these approaches rely (with the exception of [74]) on the fact that the tracking

*Cell Tracking Challenge: <http://celltrackingchallenge.net> (accessed 03/2021).

problem is inherent to the segmentation problem (“*tracking-by-detection methods*” [98]; see also [87]). That is, the key assumption made by many of these algorithms is that cells belonging to the same lineage overlap across frames (see also [20]). In this case, cell-overlap can serve as a good proxy for cell-tracking [98]. We note that in our data we cannot guarantee that the frame rate is sufficiently high for this assumption to hold.

[73, 69, 36] exploited machine learning techniques for segmentation *and* motion tracking. One key challenge here is to provide adequate training data for these methods to be successful. We here describe simulation-based techniques that can be extended to produce training data, which we use for parameter tuning [94].

The works that are most similar to ours are [17, 74, 92]. Similar to our approach, they perform a local search to identify the best cell-tracking candidates across frames. One key difference across these works are the matching criteria. Moreover, [17, 74] employ a local greedy-search, whereas we consider stochastic neural network dynamics for optimization. [92] construct score matrices within a score based neighborhood tracking method; an integer programming method is used to generate frame-to-frame correspondences between cells and the lineage map. Other approaches that consider linear programming to maximize an association score function for cell tracking can be found in [21, 20, 98]

1.2. Contributions. For image segmentation, we first apply two well-known, powerful variational segmentation algorithms to generate a large training set of correctly delineated single cells. We can then train a CNN dedicated to segmenting out each single cell. Using a CNN significantly reduces the runtime of our computational framework for cell identification. The frame-to-frame tracking of individual cells in tightly packed colonies is a significantly more challenging task, and is hence the main topic discussed in the present work. We develop a set of innovative automatic cell tracking algorithms based on the successive minimization of three dedicated cost functionals. For each pair of successive image frames, minimizing these cost functionals over all potential cell registration mappings poses significant computational and mathematical challenges. Standard gradient descent algorithms are inefficient for these discrete and highly combinatorial minimization problems. Instead, we implement the stochastic neural network dynamics of Boltzmann machines (BM), with architectures and energy functions tailored to effectively solve our combinatorial tracking problem. Our major contributions are: **i)** The design of a multi-stage cell tracking algorithm, that starts with a parent-children pairing step, followed by removal of identified parent-children triplets, and concludes with a cell to cell registration step. **ii)** The design of dedicated BM architectures, with several energy functions, respectively, minimized by true parent-children pairing and by true cell-to-cell registration. Energy minimizations are then implemented by simulation of BM stochastic dynamics. **iii)** The development of automatic algorithms for the estimation of unknown weight parameters of our BM energy functions, using convex-concave programming tools [2, 30, 81, 80]. **iv)** The evaluation of our methodology on synthetic and real image sequences of cell colonies. The massive effort involved in human expert annotation of cell colony recordings limits the availability of “*ground truth tracking*” data for dense bacterial colonies. We therefore first validated the accuracy of our cell tracking algorithms on recordings of simulated cell colonies, generated by the dedicated cell colony simulation software [94]. This provided us with ground truth frame-by-frame registration for cell lineages, enabling us to validate our methodology.

1.3. Outline. In Sec. 2 we describe the synthetic image sequence of cell colonies considered here as benchmarks for our cell tracking algorithms. In Sec. 3 we describe key cell characteristics involved in our cost functionals. We define valid cell registration mappings between successive image frames in Sec. 4. We outline how to automatically calibrate the weights of our various penalty terms in Sec. 5. Our algorithms for pairing parent cells with their children and for cell-to-cell registration are developed in sections Sec. 6 through Sec. 12. We present our main validation results on long image sequences (time series of images) in Sec. 13 and conclude with Sec. 14.

2. Benchmark: Synthetic Videos of Simulated Cell Colonies. To validate our cell tracking algorithms, we consider simulated image sequences of dense cell populations. We refer to [94] for a detailed description of this mathematical model and its implementation. The simulated cell colony dynamics are driven by an agent based model [94], which emulates live colonies of growing, moving, and dividing rod-like *E. coli* cells in a 2D microfluidic trap environment. Between two successive frames J, J_+ , cells are allowed to move until they nearly bump into each other, and to grow at multiplicative rate denoted $g.rate$ with an average value of 1.05 per minute (plus/minus a small random perturbation). For a cell of length L_0 at birth, cell division occurs at length $L_{div} = 2L_0 + \varepsilon$, where ε is a small uniformly distributed random variable. When a bacterial cell b of length L_{div} divides into two cells b_1 and b_2 , their lengths L_1, L_2 satisfy $L_1 + L_2 = L_{div}$ and L_1/L_{div} is a random number in $[0.45, 0.55]$. All these random variables are independent of each other.

The simulation keeps track of cell lineage, cell size, and cell location (among other parameters). The main output of each such simulation considered here is a binary image sequence of the cell colony with a fixed interframe duration. Each such synthetic image sequence is used as the sole input to our cell tracking algorithm. The remaining meta-data

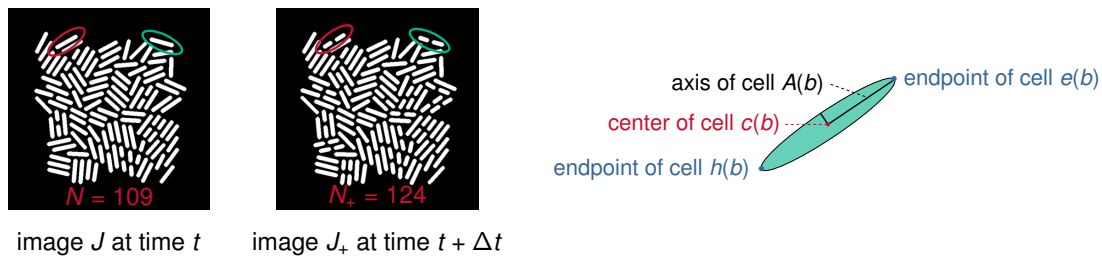


FIG. 2. **Simulated data and cell characteristics.** Left: Two successive images generated by dynamic simulation for a colony of rod-shaped bacteria. Left image J displays $N = 109$ cells at time t . At time $t + \Delta t$ with $\Delta t = 1$ min, cells have moved, grown, and some have divided. These cells are displayed in image J_+ , which contains $N_+ = 124$ cells. We highlight two cells that have undergone a division between the frames (red and green ellipses). Right: Geometry of a rod shaped bacterium. Each cell b is identified by its center $c(b)$, its long axis $A(b)$, and the two endpoints $e(b)$, $h(b)$ of $A(b)$.

generated by the simulations are only used as ground truth to evaluate the performance of our tracking algorithms.

We consider two benchmark datasets, BENCH1 and BENCH6, of several *synthetic* image sequences of simulated cell colonies with *cell growth factor* $g.rate = 1.05$ per minute. The generated binary images are of size 600×600 pixels, with interframe durations of 1 minute for BENCH1, and of 6 minutes for BENCH6. The associated image sequences involve 100 to 500 frames each. In Fig. 2 we display an example of two simulated consecutive frames separated by 1 minute. To simplify our presentation and validation tests, we control our simulations to make sure that cells will not exit the region of interest from one frame to the next, and we exclude cells that are only partially visible in the current frames

3. Cell Characteristics. We next discuss characteristics of the *E. coli* bacteria important for our tracking algorithm.

Cell Geometry. In accordance with the dynamics of bacterial colonies in microfluidic traps, the dynamic simulation software generates colonies of rod-shaped bacteria. Cell shapes can be approximated by long and thin ellipsoids, which are geometrically well identified by their center, their long axis, and the two endpoints of this long axis. The center $c(b)$ is the centroid of all pixels belonging to cell b . The long axis $A(b)$ of cell b is computed by principal component analysis (PCA). The endpoints $e(b)$ and $h(b)$ of cell b are the first and last cell pixels nearest to $A(b)$; see Fig. 2 (right) for a schematic illustration.

Cell Neighbors. For each image frame J , denote $B = B(J)$ the set of fully visible cells in J , and by $N = N(J) = \text{card}(B)$ the number of these cells. Let V be the set of all cell centers $c(b)$ with $b \in B$. Denote $delV$ the Delaunay triangulation [83] of the finite planar set V with N vertices. We say that two cells b_1, b_2 in B are *neighbors* if they verify the following three conditions:

1. (b_1, b_2) are connected by the edge $edge$ of one triangle in $delV$.
2. The edge $edge$ does not intersect any other cell in B .
3. Their centers verify $\|c(b_1) - c(b_2)\| \leq \rho$, where $\rho > 0$ is a user defined parameter.

For the synthetic images of size 600×600 that we considered (see Sec. 2), we take $\rho = 80$ pixels. We write $b_1 \sim b_2$ for short, whenever b_1, b_2 are neighbors (i.e., satisfy the three conditions identified above).

Cell Motion. Let J, J_+ denote two successive images (i.e., frames). Denote $B = B(J)$, $B_+ = B(J_+)$ the associated sets of cells. Superpose temporarily the images J and J_+ so that they then have the same center pixel. Any cell $b \in B$, which does not divide in the interframe $J \rightarrow J_+$, becomes a cell b_+ in image J_+ . The “motion vector” of cell b from frame J to J_+ is then defined by $v(b) = c(b_+) - c(b)$. If the cell b does divide between J and J_+ , denote b_{div} the last position reached by cell b at the time of cell division, and define similarly the motion $v(b) = c(b_{div}) - c(b)$. In our experimental recordings of real bacterial colonies with interframe duration 6 min, there is a *fixed number* $w > 0$ such that $\|v(b)\| \leq w/2$ for all cells $b \in B(J)$ for all pairs J, J_+ . In particular, we observed that for real image sequences, $w = 100$ pixels is an adequate choice. Consequently, we select $w = 100$ pixels for all simulated image sequences of BENCH6. For BENCH1 we select $w = 45$ pixels, again based on a comparison with real experimental recordings. Overall, the meta-parameter w is assumed to be a fixed number and to be known, since $w/2$ is an observable upper bound for the cell motion norm for a particular image sequence of a lab experiment.

Target Window. Recall that J, J_+ are temporarily superposed. Let $U(b) \subset J_+$ be a square window of width w , with the same center as cell b . The *target window* $W(b)$ is the set of all cells in B_+ having their centers in $U(b)$. Since

$\|v(b)\| \leq w/2$, the cell b_+ must belong to the target window $W(b) \subset B_+$.

4. Registration Mappings. Next we discuss our assumptions on a valid registration mapping that establishes cell-to-cell correspondences between two frames. Let J, J_+ denote two successive images, with cell sets B and B_+ , respectively. As above, we let $N = \text{card}(B)$, and $N_+ = \text{card}(B_+)$. Our goal is to track each cell from J to J_+ . For each cell $b \in B$, there exist three possible evolutions between J and J_+ :

Case 1: Cell $b \in B$ did **not** divide in the interframe $J \rightarrow J_+$, and has become a cell $f(b) \in B_+$; that is, $f(b)$ has grown and moved during the interframe time interval.

Case 2: Cell $b \in B$ divided between J and J_+ , and generated two children cells $b_1, b_2 \in B_+$; we then denote $f(b) = (b_1, b_2) \in B_+ \times B_+$.

Case 3: Cell $b \in B$ disappeared in the interframe $J \rightarrow J_+$, so that $f(b)$ is not defined.

To simplify our exposition, we *ignore Case 3*, which can be handled by minor extensions of the cost minimization algorithms develop below. Consequently, a valid (true) registration mapping f will take values in the set $\{B_+\} \cup \{B_+ \times B_+\}$.

5. Calibration of Cost Function Weights. With the notation we introduced, fix any two finite sets A, A_+ . Let $G := \{g : A \rightarrow A_+\}$ be the set of all mappings $g : A \rightarrow A_+$. Fix m penalty functions $\text{pen}_k(g) \geq 0, k = 1, \dots, m$. Let $g^* \in G$ be the ground truth mapping we want to discover through minimization in g of some given cost function $\text{COST}(g)$ defined by the linear combination of the penalty functions $\text{pen}_k(g)$, the contributions of which are controlled by the cost function weights $\lambda_k > 0$. In this section, we present a generic *weight calibration algorithm*, extending a technique introduced and applied in [11, 12] for Markov random fields based image analysis.

The cost function must perform well (with the same weights) for hundreds of pairs of (synthetic) images J, J^+ . We consider one such synthetic pair for which the ground truth registration mapping $f \in G$ is known, and use it to compute an adequate set of weights, which will then be used on all other synthetic pairs J, J^+ . Notice, that for experimental recordings of real cell colonies, no ground truth registration mappings f is available. In this case, f should be replaced by a set of user constructed, correct *partial* mappings defined on small subsets of A . The proposed weight calibration algorithm will also work in those situations.

We now show how knowing one ground truth mapping f can be used to derive the best feasible weights ensuring that f should be a plausible minimizer of the cost functional $\text{COST}(g)$ over $g \in G$. Let $\text{PEN}(g) = [\text{pen}_1(g), \dots, \text{pen}_m(g)]$ be the vector of m penalties for any mapping $g \in G$. Let $\Lambda = [\lambda_1, \dots, \lambda_m]$ be the weights vectors. Then, $\text{COST}(g) = \langle \Lambda, \text{PEN}(g) \rangle$. Replacing g by another mapping $h \neq g$ induces the penalty changes $\Delta \text{PEN}_{g,h} = \text{PEN}(h) - \text{PEN}(g)$ and the cost change $\Delta \text{COST}(g, h) = \langle \Lambda, \Delta \text{PEN}_{g,h} \rangle$. Now, fix any known ground truth mapping $f \in G$. We want f to be a minimizer of COST , so we should have

$$\Delta \text{COST}(f, f') \geq 0 \quad \text{for all modifications } f \rightarrow f' \in G.$$

For each $a \in A$, select an arbitrary $s(a) \in W(a)$ (where $W(a)$ is the target window for cell a ; see Sec. 3), to define a new mapping $f' = f'_a$ from A to A_+ by $f'_a(a) = s(a)$, and $f'_a(x) \equiv f(x)$ for all $x \neq a$. Since f is a minimizer of COST , this single point modification $f \rightarrow f'_a$ must generate the following cost increase

$$\langle \Lambda, \Delta \text{PEN}(f, f'_a) \rangle = \Delta \text{COST}(f, f'_a) \geq 0.$$

Denote $V_a \in \mathbb{R}^m$ the vector $V_a = \Delta \text{PEN}(f, f'_a)$. Then, the positive vector $\Lambda \in \mathbb{R}^m, \Lambda \succeq 0$, should verify the set of linear constraints $\langle \Lambda, V_a \rangle \geq 0$ for all $a \in A$. There may be too many such linear constraints. Consequently, we *relax* these constraints by introducing a vector $y = [y(a)] \in \mathbb{R}^{\text{card}(A)}, y \succeq 0$, of slack variables $y(a) \geq 0$ indexed by all the $a \in A$. (In optimization slack variables are introduced as additional unknowns to transform inequality constraints to an equality constraint and a non-negativity constraint on the slack variables.) We require the unknown positive vector Λ and the slack variables vector y to verify the system of linear constraints:

$$\begin{aligned} (1) \quad & \langle \Lambda, V_a \rangle + y(a) = 0 \quad \text{for all } a \in A \\ & \Lambda \succeq 0, y \succeq 0 \\ & \langle \Lambda, Z \rangle \leq 1000 \end{aligned}$$

where $Z = [1, \dots, 1] \in \mathbb{R}^m$. The normalizing constant 1000 can be arbitrarily changed by rescaling. We seek high positive values for $\Delta \text{COST}(f, f'_a)$ and small L_1 -norm for the slack variable vector y . So, we will seek two vectors

$\Lambda \in \mathbb{R}^m$ and $y \in \mathbb{R}^{\text{card}(A)}$ solving the following *convex-concave* minimization problem, where $\gamma > 0$ is a user selected (large) meta parameter:

$$(2) \quad \underset{\Lambda, y}{\text{minimize}} \quad \gamma \|y\|_{L1} - \sum_{a \in A} [\langle \Lambda, V_a \rangle]^+ \quad \text{subject to (1),}$$

where we denote $[x]^+ := \max(x, 0)$ for arbitrary x . To numerically solve the constrained minimization problem (2), we use the libraries CVXPY and DCCP (disciplined convex-concave programming) [2, 30, 80, 81]. DCCP is a package for convex-concave programming designed to solve non-convex problems. It can handle objective functions and constraints with any known curvature as defined by the rules of disciplined convex programming [24]. We give examples of numerically computed weight vectors Λ below. The computing time was less than 30 seconds for the data that we have prepared. For simplicity, we just considered one step changes in our computations, which make the overlap penalty weak. To increase the accuracy of the model it is possible to consider a larger number of samples (i.e., multi-step changes). Note that the solutions Λ of (2) are of course not unique, even after normalization by rescaling.

6. Cell Divisions and Children Pairing. In this section, we present the methodology for detecting cell divisions and our approach for pairing children with their parent cells.

6.1. Number $DIV(B, B_+)$ of Cell Divisions. Fix two successive synthetic image frames J, J_+ with short inter-frame time equal to 1 minute. Their cell sets B, B_+ have cardinality N and N_+ , respectively. We assume, for ease of presentation, that all cells $b \in B$ still exist in B_+ , either as whole cells or divided into two children cells (i.e., no cells exit the field of view). This implies $N_+ \geq N$, and $DIV(B, B_+) = N_+ - N$ is the number DIV of cell divisions occurring in the interframe $J \rightarrow J_+$.

Whenever $DIV > 0$, we want to compute the unknown set $trueCH$ of true children pairs $(b_1, b_2) \in B_+ \times B_+$. Each such pair is born from the division of some unknown parent cell $b = \text{parent}(b_1, b_2)$. For the synthetic image sequence we should have $\text{card}(trueCH) = DIV$, but for computational advantage below, whenever $DIV \geq 2$ we *relax* this rigid constraint to the more pragmatic form $|\text{card}(trueCH) - DIV| \leq 1$. For realistic experimental recordings, the relaxation bound is linked to the numbers of new cells entering J_+ and of old cells exiting J_+ .

6.2. Linking Paired Children Cells to Parent Cells. For successive synthetic images J, J_+ with 1 minute interframe such that $DIV(B, B_+) > 0$, we call PCH the set of *plausible children pairs* defined as

$$(3) \quad PCH = \{(b_1, b_2) \in B_+ \times B_+ \text{ with centers } c_1, c_2 \text{ verifying } \|c_1 - c_2\| < \tau\},$$

where the threshold $\tau > 0$ is user selected and fixed for the whole benchmark set BENCH1 of synthetic image sequences.

To evaluate if a pair of cells $(b_1, b_2) \in PCH$ can qualify as a pair of children generated by division of a parent cell $b \in B$, we now quantify the “*geometric distortion*” between b and (b_1, b_2) . Cell division of b into $b_1, b_2 \in B_+$ occurs with small motion of b_1, b_2 . During the short interframe duration the initial centers c_1, c_2 of b_1, b_2 in image J move by at most $w/2$ pixels each (see Sec. 3), and their initial distance to the center c of b is roughly at most $\|A(b)\|/4$, where $A(b)$ is the long axis of cell b . Hence, the centers c, c_1, c_2 of b, b_1, b_2 should verify the constraint

$$(4) \quad \max\{\|c_1 - c\|, \|c_2 - c\|\} \leq w + \|A\|/4.$$

Define the set $SHLIN$ of potential *short lineages* as the set all triplets (b, b_1, b_2) with $b \in B, (b_1, b_2) \in PCH$, verifying the preceding constraint (4). For each potential lineage $(b, b_1, b_2) \in SHLIN$, define three terms penalizing the geometric distortions between a parent $b \in B$ and a pair of children $(b_1, b_2) \in PCH$ by the following formulas, where we denote c, c_1, c_2 the centers of cells b, b_1, b_2 and A, A_1, A_2 their long axes

1. center distortion: $\text{cen}(b, b_1, b_2) = \|c - (c_1 + c_2)/2\|$,
2. size distortion: $\text{siz}(b, b_1, b_2) = \|\|A\| - (\|A_1\| + \|A_2\|)\|$,
3. angle distortion: $\text{ang}(b, b_1, b_2) = \text{angle}(A, A_1) + \text{angle}(A, A_2) + \text{angle}(A, c_2 - c_1)$.

Here, angle denotes “*angles between non-oriented straight lines*,” and range from 0 to $\pi/2$.

Introduce three positive weights $\lambda_{\text{cen}}, \lambda_{\text{siz}}, \lambda_{\text{ang}}$ (to be estimated), and for every short lineage $(b, b_1, b_2) \in SHLIN$ define its *distortion cost* by:

$$(5) \quad \text{distortion}(b, b_1, b_2) = \lambda_{\text{cen}} \text{cen}(b, b_1, b_2) + \lambda_{\text{siz}} \text{siz}(b, b_1, b_2) + \lambda_{\text{ang}} \text{ang}(b, b_1, b_2).$$

6.3. Estimating Most Likely Parent Cell for Children Pair. For each plausible pair of children $(b_1, b_2) \in PCH$, we will compute the most likely *parent cell* $b^* = \text{parent}(b_1, b_2)$ as the cell $b^* \in B$ minimizing $\text{distortion}(b, b_1, b_2)$ in (5) over all $b \in B$, as summarized by the formula

$$(6) \quad b^* = \text{parent}(b_1, b_2) = \underset{\{b \in B \mid (b, b_1, b_2) \in SHLIN\}}{\text{argmin}} \quad \text{distortion}(b, b_1, b_2)$$

To force this minimization to yield a reliable estimate of $b^* = \text{parent}(b_1, b_2)$ for most true pairs of children (b_1, b_2) , we calibrate the weights λ_j , $j \in \{\text{cen}, \text{siz}, \text{ang}\}$ by the algorithm outlined in section Sec. 5, using as “ground truth set” a fairly small set of visually identified true short lineages (*parent, children*).

For fixed (b_1, b_2) , the set of potential parent cells $b \in B$ has very *small size* due to the constraint (4). Hence, brute force minimization of the functional $\text{distortion}(b, b_1, b_2)$ in (5) over all $b \in B$ such that $(b, b_1, b_2) \in SHLIN$, is a *fast computation* for each (b_1, b_2) in PCH . The distortion minimizing $b = b^*$ yields the most likely parent cell $\text{parent}(b_1, b_2) = b^*$.

6.4. Penalties to Control Children Pair Matching. True pairs of children cells $pch = (b_1, b_2) \in PCH$ must verify lineage and geometric constraints which we quantify via several penalties.

“*Lineage*” Penalty.. Valid children pairs $(b_1, b_2) \in PCH$ should be correctly matchable with their most likely parent cell $b^* = \text{parent}(b_1, b_2)$ (see (6)). So, we define the “*lineage*” penalty $\text{lin}(b_1, b_2) = \text{distortion}(b^*, b_1, b_2)$ by

$$\text{lin}(b_1, b_2) = \underset{\{b \in B \mid (b, b_1, b_2) \in SHLIN\}}{\text{argmin}} \quad \text{distortion}(b, b_1, b_2) = \text{distortion}(\text{parent}(b_1, b_2), b_1, b_2).$$

Notice that the computation of $\text{lin}(b_1, b_2)$ is quite fast.

“*Gap*” Penalty.. Denote $\text{tips}(b)$ the set of two endpoints of any cell b . For any pair $pch = (b_1, b_2) \in PCH$, define endpoints $x_1 \in \text{tips}(b_1)$, $x_2 \in \text{tips}(b_2)$ and the *gap* penalty $\text{gap}(b_1, b_2)$ by

$$(7) \quad \text{gap}(b_1, b_2) = \|x_1 - x_2\| = \min\{\|x - y\| \text{ for } (x, y) \in TIPS\}$$

with $TIPS = \text{tips}(b_1) \times \text{tips}(b_2)$.

“*Dev*” Penalty.. For rod-shaped bacteria, a true pair $(b_1, b_2) \in PCH$ of just born children must have a small $\text{gap}(b_1, b_2) = \|x_1 - x_2\|$ and roughly aligned cells b_1 and b_2 . For $(b_1, b_2) \in PCH$, we quantify the *deviation from alignment* $\text{dev}(b_1, b_2)$ as follows. Let x_1, x_2 be the closest endpoints of b_1, b_2 (see (7)). Let str_{12} be the straight line linking the centers c_1, c_2 of b_1, b_2 . Let d_1, d_2 be the distances from x_1, x_2 to the line str_{12} . Set then

$$\text{dev}(b_1, b_2) = \frac{d_1 + d_2}{\|c_2 - c_1\|}.$$

“*Ratio*” Penalty.. True children pairs must have nearly equal lengths. So, for $(b_1, b_2) \in PCH$ with lengths L_1, L_2 we define the length *ratio penalty* by

$$\text{ratio}(b_1, b_2) = |(L_1/L_2) + (L_2/L_1) - 2|.$$

“*Rank*” penalty.. Let L_{\min} be the minimum cell length over all cells in B_+ . In B_+ , children pairs (b_1, b_2) just born during interframe $J \rightarrow J_+$ must have lengths L_1, L_2 close to L_{\min} . So, for $(b_1, b_2) \in PCH$, we define the *rank* penalty by

$$\text{rank}(b_1, b_2) = |(L_1/L_{\min}) - 1| + |(L_2/L_{\min}) - 1|.$$

6.5. Cost Function Dedicated to Children Pairing. Given two successive images J, J_+ with a positive number $DIV = N_+(J) - N(J)$ of cell divisions, the set $\text{trueCH} \subset B_+ \times B_+$ of true children pairs will have size DIV and be a subset of the set PCH of plausible children pairs. In the search for trueCH , the unknown is a subset X of PCH . We build a cost function $\phi(X)$ approximately minimized when X is close to trueCH . The main term of $\phi(X)$ will be the sum over all pairs in X of a weighted linear combination of the penalty functions $\{\text{lin}, \text{gap}, \text{dev}, \text{ratio}, \text{rank}\}$. Another penalty will ensure that X contains no overlapping pairs. The minimization of $\phi(X)$, as well as of several further combinatorial cost functions below will be implemented by intensive simulations of BMs. We present these stochastic neural networks, next.

7. Generic Boltzmann Machines (BMs). Minimization of our main cost functionals is a heavily combinatorial task, since the unknown variable is a mapping between two finite sets of sizes ranging from 80 to 120. To handle these minimizations, we use BMs originally introduced by Hinton et al. (see [1, 38]). Indeed, these recurrent stochastic neural networks can efficiently emulate some forms of simulated annealing.

Each BM implemented here is a network $BM = \{U_1, \dots, U_N\}$ of N *stochastic neurons* U_j . At time $t = 0, 1, 2, \dots$, each neuron U_j has a random state $Z_j(t)$ belonging to a fixed finite set $W(j)$. The *configuration* $Z(t) = \{Z_1(t), \dots, Z_N(t)\}$ of the whole network BM thus belongs to the *configurations set* $CONF = W(1) \times \dots \times W(N)$. Neurons interactivity is specified by a finite set CLQ of *cliques*. Each clique K is a subset of $S = \{1, \dots, N\}$. During configuration updates $Z(t) \rightarrow Z(t+1)$, neurons may interact only if they are in the same clique. Here, all cliques K are of small sizes 1, or 2, or 3.

For each clique K , one specifies an energy function $J_K(z)$ defined for all $z \in CONF$, with $J_K(z)$ depending only on the z_j such that $j \in K$. The full energy $E(z)$ of configuration z is then defined by

$$E(z) = \sum_{K \in CLQ} J_K(z).$$

The BM stochastic dynamics $Z(t) \rightarrow Z(t+1)$ is driven by the energy function $E(z)$, and by a fixed decreasing sequence of *virtual temperatures* $Temp(t) > 0$, tending slowly to 0 as $t \rightarrow \infty$. Here we use standard temperature schemes of the form $Temp(t) \equiv c\eta^t$ with fixed $c > 0$ and slow *decay rate* $0.99 < \eta < 1$.

We have implemented the classical “asynchronous” BM dynamics. At each time t , *only one* random neuron U_j may modify its state, after reading the states of all neurons belonging to cliques containing U_j . A much faster alternative, implementable on GPUs, is the “synchronous” BM dynamics, where at each time t roughly 50% of all neurons may simultaneously modify their states (see [10, 9, 14]). The detailed BM dynamics is presented in the appendix (see Sec. A).

Recall that when the virtual temperatures $Temp(t)$ decrease slowly enough to 0, the energy $E(Z(t))$ converges in probability to a local minimum of the BM energy $E(z)$ over all configurations $z \in CONF$.

8. Optimized Children Pairing. Next, we present a formulation of an optimization problem to pair parents with their children. Fix successive images J, J_+ with a positive number of cell divisions $DIV = N_+ - N$. Denote $PCH = \{pch_1, pch_2, \dots, pch_m\}$ the set of m plausible children pairs (b_1, b_2) in B_+ . The penalties *lin*, *gap*, *dev*, *ratio*, and *rank* defined above for all pairs $(b_1, b_2) \in PCH$ determine five numerical vectors $LIN, GAP, DEV, RAT, RANK$ in \mathbb{R}^m with coordinates $LIN_j = \text{lin}(pch_j)$, $GAP_j = \text{gap}(pch_j)$, $DEV_j = \text{dev}(pch_j)$, $RAT_j = \text{ratio}(pch_j)$, $RANK_j = \text{rank}(pch_j)$.

We now define a *binary* BM constituted by m *binary* stochastic neurons U_j , $j = 1 \dots m$. At time $t = 0, 1, 2, \dots$, each U_j has a random *binary valued state* $Z_j(t) = 1$ or 0. The random configuration $Z(t) = [Z_1(t), \dots, Z_m(t)]$ of this BM belongs to the configuration space $CONF = \{0, 1\}^m$ of all binary vectors $z = [z_1, \dots, z_m]$. Let SUB be the set of all subsets of PCH . Each configuration $z \in CONF$ is the indicator function of a subset $sub(z)$ of PCH . We view each $sub(z) \in SUB$ as a possible estimate for the unknown set $trueCH \subset B_+ \times B_+$ of true children pairs (b_1, b_2) . For each potential estimate $sub(z)$ of $trueCH$, the “lack of quality” of the estimate $sub(z)$ will be penalized by the *energy function* $E(z) \geq 0$ of our binary BM. We now specify the energy $E(z)$ for all $z \in CONF$ by combining the penalty terms introduced above.

True children pairs born from distinct parents must clearly not intersect. To enforce this constraint, define the symmetric $m \times m$ binary matrix $[Q_{j,k}]$ by **i**) $Q_{j,k} = 1$ if $j \neq k$ and the two pairs pch_j, pch_k have **one** cell in common, **ii**) $Q_{j,k} = 0$ if $j \neq k$ and the two pairs pch_j, pch_k have **no** cell in common, **iii**) $Q_{j,j} = 0$ for all j .

The quadratic penalty $z \mapsto \langle z, Qz \rangle$ is non-negative for $z \in CONF$, and must be zero if $sub(z) = trueCH$. Introduce six positive weight parameters to be selected further on λ_j , $j \in \{\text{lin}, \text{gap}, \text{dev}, \text{rat}, \text{rank}, Q\}$. Define the vector $V \in \mathbb{R}^m$ as a weighted linear combination of the penalty vectors $LIN, GAP, DEV, RAT, RANK$

$$V = \lambda_{\text{lin}} LIN + \lambda_{\text{gap}} GAP + \lambda_{\text{dev}} DEV + \lambda_{\text{rat}} RAT + \lambda_{\text{rank}} RANK.$$

For any configuration $z \in CONF$, the BM energy $E(z)$ is defined by the *quadratic function*

$$E(z) = \langle V, z \rangle + \lambda_Q \langle z, Qz \rangle$$

We already know that the unknown set $trueCH$ of true children pairs must have cardinal $DIV = N_+ - N$. So we seek a configuration $z^* \in CONF$ minimizing the energy $E(z)$ under the rigid constraint $\text{card}\{sub(z)\} = DIV$. Let $ONE \in \mathbb{R}^m$ be the vector with all its coordinates equal to 1. The constraint on z can be reformulated as $\langle ONE, z \rangle = DIV$. We want the unknown $trueCH$ to be close to the solution z^* of the constrained minimization problem

$$z^* = \underset{z \in CONF}{\text{argmin}} E(z) \quad \text{subject to} \quad \langle ONE, z \rangle = DIV$$

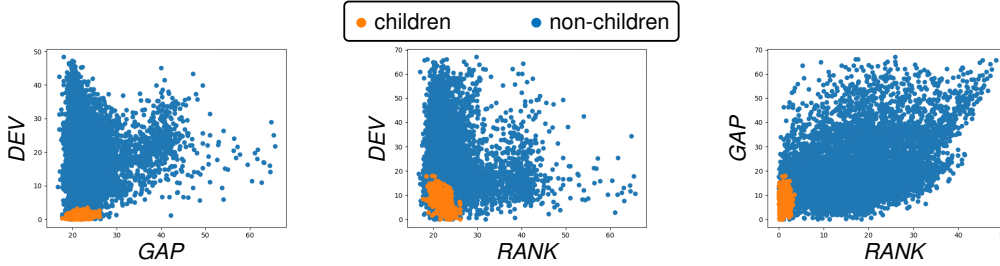


FIG. 3. *Scatter plots for tandems of the penalty terms DEV, GAP, and RANK.* We mark in orange the true children pairs and in blue invalid children pairs. These plots allow us to identify appropriate empirical thresholds to trim the (considered synthetic) data in order to reduce the computational complexity of the parent-children pairing.

To force this minimization to yield a reliable estimate of *trueCH*, we calibrate the six weights

$$\lambda_j, j \in \{\text{lin}, \text{gap}, \text{dev}, \text{rat}, \text{rank}, Q\}$$

by the algorithm in Sec. 5. Denote $CONF_1$ the set of all $z \in CONF$ such that $\langle ONE, z \rangle = DIV$. To minimize $E(z)$ under the constraint $z \in CONF_1$, fix a slowly decreasing temperature scheme $Temp(t)$ as in Sec. 7. We need to force the BM stochastic configurations $Z(t)$ to remain in $CONF_1$. Then, for large time step t , the $Z(t)$ will converge in probability to a configuration $z^* \in CONF_1$ approximately minimizing $E(z)$ under the constraint $z \in CONF_1$.

Start with any $Z(0) \in CONF_1$. Assume that for $0 \leq s \leq t$, one has already dynamically generated BM configurations $Z(s) \in CONF_1$. Then, randomly select two sites j, k such that $Z_j(t) = 1$ and $Z_k(t) = 0$. Compute a virtual configuration Y by setting $Y_j = 0, Y_k = 1$, and $Y_i \equiv Z_i$ for all sites i different from j and k . Compute the energy change $\Delta E = E(Y) - E(Z(t))$, and the probability $p(t) = \exp(-D/Temp(t))$, where $D = \max\{0, \Delta E\}$. Then randomly select $Z(t+1) = Y$ or $Z(t+1) = Z(t)$ with respective probabilities $p(t)$ and $(1 - p(t))$. Clearly, this forces $Z(t+1) \in CONF_1$.

9. Performance of Automatic Children Pairing on Synthetic Videos. In the following subsections we provide experimental results for pairing children and parent cells.

9.1. Children Pairing: Fast BM simulations. For $m = \text{card}(PCH) \leq 1000$, one can reduce the computational cost for BM dynamics simulations by pre-computing and storing the $m \times m$ symmetric binary matrix Q , as well as the m -dimensional vectors $LIN, GAP, DEV, RAT, RANK$ and their linear combination V . A priori reduction of m significantly reduces the computing times, and can be implemented by trimming away the pairs $pch_j \in PCH$ for which the penalties $LIN_j, GAP_j, DEV_j, RAT_j$, and $RANK_j$ are all larger than predetermined empirical thresholds. We performed a study on 100 successive (synthetic) images. We show scatter plots for the most informative penalty terms in Fig. 3. These plots allow us to determine adequate thresholds for the penalty terms. We observed that for the synthetic and real data we considered the trimming of DEV, GAP , and $RANK$ reduced the percentage of invalid children pairs by 95%, therefore drastically reducing the combinatorial complexity of the problem.

The quadratic energy function $E(z)$ is the sum of clique energies $J_K(z)$ involving only cliques of cardinality 1 and 2. For any clique $K = \{j\}$ of cardinality 1, with $1 \leq j \leq m$, one has $J_K(z) = V_j z_j$. For any clique $K = \{j, k\}$ of cardinality 2, with $1 \leq j < k \leq m$, one has $J_K(z) = 2Q_{j,k} z_j z_k$. A key computational step when generating $Z(t+1)$ is to evaluate the energy change ΔE when one flips the binary values $Z_j(t) = 1$ and $Z_k(t) = 0$ by the new value $(1 - z_i)$ for a fixed single site i . This step is quite fast since it uses only the numbers V_j, V_k , and $\langle q(j), Z(t) \rangle, \langle q(k), Z(t) \rangle$, where $q(i)$ is the i^{th} row of the matrix Q .

9.2. Children Pairing: Implementation on Synthetic Videos. We have implemented our children pairing algorithms on synthetic image sequences having 100 to 500 image frames with 1 minutes interframe (benchmark set BENCH1; see Sec. 2). The cell motion bound $w/2$ per interframe was defined by $w = 20$ pixels. The parameter τ that defines the sets PCH of plausible children pairs (see (3)) was set at $\tau = 45$ pixels.

The known true cell registrations indicated that in our typical BENCH1 image sequence, the successive sets PCH had average cardinals of 120, while the number of true children pairs per PCH roughly ranged from 2 to 6 with a median of 4. The size of the reduced configuration space $CONF_1$ per image frame thus ranged from 10^4 to $120^6/6! = 4.2 \cdot 10^9$ with a median of $9 \cdot 10^6$.

Our weights estimation technique introduced in Sec. 5 yields the weights $[\lambda_{\text{cen}}, \lambda_{\text{siz}}, \lambda_{\text{ang}}] = [0.255, 0.05, 0.05]$ and $[\lambda_{\text{gap}}, \lambda_{\text{dev}}, \lambda_{\text{rat}}, \lambda_{\text{rank}}] = [0.01, 1, 0.0001, 0.05]$ or the penalties introduced in Sec. 13. To reduce the computing

TABLE 1

Accuracies of parent-children pairing algorithm. We applied our parent-children pairing algorithm to three long synthetic image sequences BENCH1 (500 frames), BENCH2 (300 frames), and BENCH3 (300 frames), with interframe intervals of 1, 2, 3 minutes, respectively. The table summarizes the resulting pcg-accuracies. Note that pcg-accuracies are practically always at 100%. For BENCH2 pcg-accuracies are 100% for 298 frames out of 300, and for the remaining two frames, accuracies were still high at 93% and 96%. For BENCH3 the average pcg-accuracy for the 3 minute interframe is 99%.

image sequence	pcg-accuracy	number of frames
BENCH1	$acc = 100\%$	500 out of 500
BENCH2	$acc = 100\%$	298 out of 300
BENCH2	$99\% \geq acc \geq 93\%$	2 out of 300
BENCH3	$acc = 100\%$	271 out of 300
BENCH3	$99\% \geq acc \geq 95\%$	17 out of 300
BENCH3	$94\% \geq acc \geq 90\%$	12 out of 300

time for hundreds of BM energy minimizations on the BENCH1 image sequences, we excluded obviously invalid children pairs in each PCH set, by simultaneously thresholding of the penalty terms. The BM temperature scheme was $Temp(t) = 1000(0.995)^t$, with the number of epochs capped at 5000. The average CPU time for BM energy minimization dedicated to optimized children pairing was about 30 seconds per frame.

9.3. Parent-Children Matching: Accuracy on Synthetic Videos. For each successive image pair J, J_+ , with cells B, B_+ of cardinality $N < N_+$, our parent-children matching algorithm computes a set SHL of short lineages (b, b_1, b_2) , where the cell $b \in B$ is expected to be the parent of cells $b_1, b_2 \in B_+$. Recall that $DIV = N_+ - N$ provides the number of cell divisions during the interframe $J \rightarrow J_+$. The number VAL of correctly reconstructed short lineages $(b, b_1, b_2) \in SHL$ is obtained by direct comparison to the known ground truth registration $J \rightarrow J_+$. For each frame J , we define the pcg-accuracy of our Parent-Children Pairing algorithm as the ratio VAL/DIV .

We have tested our parent-children matching algorithm on three long synthetic image sequences BENCH1 (500 frames), BENCH2 (300 frames), and BENCH3 (300 frames), with respective interframes of 1, 2, and 3 minutes. For each frame J_k , we computed the pcg-accuracy between J_k and J_{k+1} .

We report the accuracies of our parent-children pairing algorithms in Tab. 1. For BENCH1, all 500 pcg-accuracies reached 100%. For BENCH2, pcg-accuracies reach 100% for 298 frames out of 300, and for the remaining two frames, accuracies were still high at 93% and 96%. For BENCH3, where interframe duration was longest (3 minutes), the 300 pcg-accuracies decreased slightly but still averaged 99%, and never fell below 90%.

10. Reduction to Registrations with No Cell Division. Fix successive frames J, J_+ and their cell sets B, B_+ . We seek the unknown registration mapping $f: B \rightarrow \{B_+ \cup (B_+ \times B_+)\}$, where $f(b) \in B_+$ iff cell b did not divide during the interframe $J \rightarrow J_+$ and $f(b) = (b_1, b_2) \in B_+ \times B_+$ iff cell b divided into (b_1, b_2) during the interframe.

If $\text{card}(B) = N < N_+ = \text{card}(B_+)$, we know that the number of cell divisions during the interframe $J \rightarrow J_+$ should be $DIV = DIV(B, B_+) = N_+ - N > 0$. We then apply the parent-children matching algorithms outlined above to compute a set $SHL = SHL(B, B_+)$ of short lineages (b, b_1, b_2) with $b \in B, b_1, b_2 \in B_+$ and $\text{card}(SHL) = DIV$. For each $(b, b_1, b_2) \in SHL$, the cell b is computed by $b = \text{parent}(b_1, b_2)$ as the parent cell of the two children cells $b_1, b_2 \in B_+$.

For each $(b, b_1, b_2) \in SHL$, eliminate from B the parent cell, b , and eliminate from B_+ the two children cells b_1, b_2 . We are left with two residual sets, $resB \subset B$ and $resB_+ \subset B_+$, having the same cardinality, $N - DIV = N_+ - 2DIV$. Assuming that our set SHC of short lineages is correctly determined, the cells $b \in resB$ should not divide in the interframe $J \rightarrow J_+$, and hence have a single (still unknown) registration $f(b) \in resB_+$. Thus, the still unknown part of the registration f is a bijection from $resB$ to $resB_+$.

Let $divB = B - resB$ and $divB_+ = B_+ - resB_+$. For each $b \in divB$, the cell b divides into the unique pair of cells, $(b_1, b_2) \in divB_+ \times divB_+$, such that $(b, b_1, b_2) \in SHL$. Hence, we can set $f(b) = (b_1, b_2)$ for all $b \in divB$. Thus, the remaining problem to solve is to compute the bijective registration $f: resB \rightarrow resB_+$. We have reduced the registration discovery to a new problem, where *no cell divisions occur* in the interframe duration. In what follows, we present our algorithm to solve this registration problem.

11. Automatic Cell Registration after Reduction to Cases with No Cell Division. As indicated above, we can *explicitly reduce* the generic cell tracking problem to a problem where there is *no cell division*. We consider images J, J_+ with associated cell sets B, B_+ such that $N = \text{card}(B) = \text{card}(B_+)$. Hence, there are no cell divisions in the interframe $J \rightarrow J_+$ and the map f of this reduced problem is (in principle) a bijection $f: B \rightarrow B_+$ with $\text{card}(B) = \text{card}(B_+)$. We show two typical successive images we use for testing with no cell division generated by the simulation software [94]

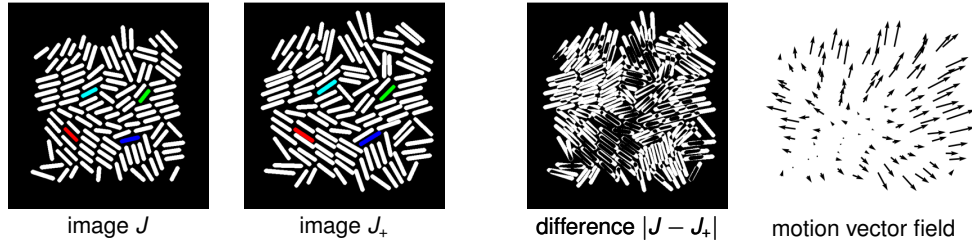


FIG. 4. **Simulated cell dynamics.** From left to right, two successive simulated images J and J_+ with an interframe time of six minutes and no cell division, their image difference $|J - J_+|$, and the associated motion vectors. For the image J and J_+ we color four pairs of cells in $B \times B_+$, which should be matched by the true cell registration mapping. Notice that the motion for an interframe time of six minutes is significant. We can observe that even without considering cell division, we can no longer assume that corresponding cells in frame J and J_+ overlap.

(see Sec. 2) in Fig. 4.

11.1. The set MAP of Many-to-One Cell Registrations. We have reduced the registration search to a situation where during the interframe $J \rightarrow J_+$, no cell has divided, no cell has disappeared, and no cell has suddenly emerged in B_+ without originating from B . The unknown registration $f : B \rightarrow B_+$ should then in principle be injective and onto. However, for computational efficiency, we will temporarily relax the bijectivity constraint on f . We will seek f in the set MAP of all *many-to-one mappings* $f : B \rightarrow B_+$ such that for each $b \in B$, the cell $f(b)$ is in the target window $W(b) \subset B_+$ (see section Sec. 3).

11.2. Registration Cost Functional. To design a cost functional $\text{cost}(f)$, which should be roughly minimized when $f \in MAP$ is very close to the true registration from B to B_+ , we linearly combine penalties $\text{match}(f)$, $\text{over}(f)$, $\text{stab}(f)$, $\text{flip}(f)$ weighted by unknown positive weights λ_{match} , λ_{over} , λ_{stab} , λ_{flip} , to write, for all registrations $f \in MAP$,

$$(8) \quad \text{cost}(f) = \lambda_{\text{match}} \text{match}(f) + \lambda_{\text{over}} \text{over}(f) + \lambda_{\text{stab}} \text{stab}(f) + \lambda_{\text{flip}} \text{flip}(f).$$

We specify the individual terms that appear in (8) below. Ideally, the minimizer of $\text{cost}(f)$ over all $f \in MAP$ is close to the unknown true registration mapping $f : B \rightarrow B_+$. To enforce a good approximation of this situation, we first estimate efficient positive weights by applying our calibration algorithm (see Sec. 5). The actual minimization of $\text{cost}(f)$ over all $f \in MAP$ is then implemented by a BM described in Sec. 12.

11.2.1. Cell Matching Likelihood: $\text{match}(f)$. Here, we extend a pseudo likelihood approach used to estimate parameters in Markov random fields modeling by Gibbs distributions (see [49]). Recall that $g.\text{rate}$ is the *known* average cell growth rate. For any cells $b \in B$, $b_+ \in B_+$, the geometric quality of the matching $b \mapsto b_+$ relies on three main characteristics: (i) motion $c(b_+) - c(b)$ of the cell center $c(b)$, (ii) angle between the long axes $A(b)$ and $A(b_+)$, (iii) cell length ratio $\|A(b_+)\|/\|A(b)\|$. So, for all $b \in B$ and b_+ in the target window $W(b)$, define

1. Kinetic energy: $\text{kin}(b, b_+) = \|c(b) - c(b_+)\|^2$.
2. Distortion of cell length:

$$\text{dis}(b, b_+) = |\log(\|A(b_+)\|/\|A(b)\|) - \log g.\text{rate}|^2.$$

3. Rotation angle: $0 \leq \text{rot}(b, b_+) \leq \pi/2$ is the geometric angle between the straight lines carrying $A(b)$ and $A(b_+)$.

Fix $b \in B$, and let b' run through the whole target window $W(b)$. The finite set of values thus reached by the kinetic penalties $\text{kin}(b, b')$ has two smallest values $\text{kin}_1(b)$, $\text{kin}_2(b)$. Define $\text{list.kin} = \bigcup_{b \in B} \{\text{kin}_1(b), \text{kin}_2(b)\}$, which is a list of $2N$ “low” kinetic penalty values. Repeat this procedure for the penalties $\text{dis}(b, b')$ and $\text{rot}(b, b')$ to similarly define a list.dis of $2N$ “low” distortion penalty values, and a list.rot of $2N$ “low” rotation penalty values.

The three sets list.kin , list.dis , list.rot can be viewed as three random samples of size $2N$, respectively, generated by three unknown probability distributions P_{kin} , P_{dis} , P_{rot} . We approximate these three probabilities by their *empirical* cumulative distribution functions CDF_{kin} , CDF_{dis} , CDF_{rot} , which can be readily computed. We now use the right tails of these three CDFs to compute separate probabilistic evaluations of how *likely* the matching of cell $b \in B$ with cell $b_+ \in W(b)$ is. For any fixed mapping $f \in MAP$, and any $b \in B$, set $b_+ = f(b)$. Compute the three penalties $v_{\text{kin}} = \text{kin}(b, b_+)$, $v_{\text{dis}} = \text{dis}(b, b_+)$, $v_{\text{rot}} = \text{rot}(b, b_+)$, and define three associated “likelihoods” for the matching

$$b \rightarrow b_+ = f(b).$$

$$\begin{aligned}\text{LIK}_{\text{kin}}(b, b_+) &= 1 - \text{CDF}_{\text{kin}}(v_{\text{kin}}), \\ \text{LIK}_{\text{dis}}(b, b_+) &= 1 - \text{CDF}_{\text{dis}}(v_{\text{dis}}), \\ \text{LIK}_{\text{rot}}(b, b_+) &= 1 - \text{CDF}_{\text{rot}}(v_{\text{rot}}).\end{aligned}$$

High values of the penalties v_{kin} , v_{dis} , v_{rot} thus will yield three small likelihoods for the matching $b \rightarrow b_+ = f(b)$. With this, we can define a “joint likelihood” $0 \leq \text{LIK}(b, b_+) \leq 1$ evaluating how likely is the matching $b \rightarrow b_+ = f(b)$:

$$(9) \quad \text{LIK}(b, b_+) = \text{LIK}_{\text{kin}}(b, b_+) \text{LIK}_{\text{dis}}(b, b_+) \text{LIK}_{\text{rot}}(b, b_+).$$

Note that higher values of $\text{LIK}(b, b_+)$ correspond to a better geometric quality for the matching of b with $b_+ = f(b)$. To avoid vanishingly small likelihoods, whenever $\text{LIK}(b, b_+) < 1e-6$, we replace it by $1e-6$. Then, for any mapping $f \in \text{MAP}$, we define its *likelihood* $\text{lik}(f)$ by the finite product

$$\text{lik}(f) = \prod_{b \in B} \text{LIK}(b, f(b)).$$

The product of these N likelihoods is typically very small, since $N = \text{card}(B)$ can be large. So, we evaluate the geometric matching quality $\text{match}(f)$ of the mapping f via the averaged *log-likelihood* of f , namely,

$$\text{match}(f) = -\frac{1}{N} \log \text{lik}(f) = -\frac{1}{N} \sum_{b \in B} \log \text{LIK}(b, f(b)).$$

Good registrations $f \in \text{MAP}$ should yield small values for the criterion $\text{match}(f)$.

11.2.2. Overlap: $\text{over}(f)$. We expect *bona fide* cell registrations $f \in \text{MAP}$ to be bijections. Consequently, we want to penalize mappings f which are many-to-one. We say that two distinct cells $(b, b') \in B \times B$ do *overlap* for the mapping $f \in \text{MAP}$ if $f(b) = f(b')$. The total number of overlapping pairs (b, b') for f defines the *overlap penalty*:

$$\text{over}(f) = \frac{1}{\text{card}(B)} \sum_{b \in B} \sum_{b' \in B} 1_{f(b)=f(b')}.$$

11.2.3. Neighbor Stability: $\text{stab}(f)$. Let $B = \{b_1, \dots, b_N\}$. Denote G_i the set of all neighbors for cell b_i in B (i.e., $b_j \sim b_i \iff b_j \in G_i$; see Sec. 3). For *bona fide* registrations $f \in \text{MAP}$, and for most pairs of neighbors $b_i \sim b_j$ in B , we expect $f(b_i)$ and $f(b_j)$ to remain neighbors in B_+ . Consequently, we penalize the lack of “neighbors stability” for f by

$$\text{stab}(f) = \sum_i \sum_{j \neq i} \frac{1}{N|G_i||G_j|} 1_{b_i \sim b_j} 1_{f(b_i) \not\sim f(b_j)}.$$

11.2.4. Neighbor Flip: $\text{flip}(f)$. Fix any mapping $f \in \text{MAP}$, any cell $b \in B$ and any two neighbors b', b'' of b in B . Let $z = f(b)$, $z' = f(b')$, $z'' = f(b'')$. Let c, c', c'' and d, d', d'' be the centers of cells b, b', b'' and z, z', z'' . Let α be the oriented angle between $c' - c$ and $c'' - c$, and let α_f be the angle between $d' - d$ and $d'' - d$, respectively. We say that the mapping f has *flipped* cells b', b'' around b , and we set $\text{FLIP}(f, b, b', b'') = 1$ if z', z'' are both neighbors of z , and the two angles α, α_f have *opposite signs*. In all other cases, we set $\text{FLIP}(f, b, b', b'') = 0$.

For any registration $f \in \text{MAP}$, define the *flip penalty* for f by

$$\text{flip}(f) = \sum_{b \in B} \sum_{b' \in B} \sum_{b'' \in B} \frac{1}{N|G(b)|^2} \text{FLIP}(f, b, b', b''),$$

where $G(b)$ is the neighborhood of cell b in B . In Fig. 5 we illustrate an example of an unwanted cell flip.

12. BM Minimization of Registration Cost Function. In what follows, we define the optimization problem for the registration of cells from one frame to another (i.e., cell tracking), as well as associated methodology and parameter estimates.

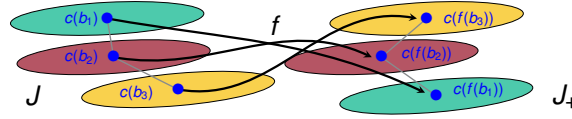


FIG. 5. *Illustration of an undesirable flip for the mapping f . The cells b_1 and b_3 are neighbors of b_2 , and mapped by f on neighbors $z_1 = f(b_1), z_3 = f(b_3)$ of $z_2 = f(b_2)$, as should be expected for bona fide cells registrations. But for this mapping f , we have z_3 above z_2 above z_1 , whereas for the original cells we had b_1 above b_2 above b_3 . Our cost function penalizes flips of this nature.*

12.1. BM Minimization of $\text{cost}(f)$ over $f \in \text{MAP}$. Let B, B_+ be two successive sets of cells. As outlined above, we have reduced the problem to one in which we can assume that $N = \text{card}(B) = \text{card}(B_+)$, so that there is no cell division during the interframe. Write $B = \{b_1, \dots, b_N\}$. For short, denote $W(j) \subset B_+$ instead of $W(b_j)$ the target window of cell b_j . We seek to minimize $\text{cost}(f)$ over all registrations $f \in \text{MAP}$. Let BM be a BM with sites $S = \{1, \dots, N\}$ and stochastic neurons $\{U_1, \dots, U_N\}$. At time t , the random state $Z_j(t)$ of U_j will be some cell z_j belonging to the target window $W(j)$ and the random configuration $Z(t) = \{Z_1(t), \dots, Z_N(t)\}$ of the whole BM belongs to the configurations set $\text{CONF} = W(1) \times \dots \times W(N)$.

To any configuration $z = \{z_1, \dots, z_N\} \in \text{CONF}$, we associate a unique cell registration $f \in \text{MAP}$ defined by $f(b_j) = z_j$ for all j , denoted by $f = \text{map}(z)$. This determines a bijection $z \mapsto f = \text{map}(z)$ from CONF onto MAP . The inverse of $\text{map} : \text{CONF} \rightarrow \text{MAP}$ will be called $\text{range} : \text{MAP} \rightarrow \text{CONF}$, and is defined by $z = \text{range}(f)$, when $z_j = f(b_j)$ for all j .

12.2. BM Energy Function $E(z)$. We now define the energy function $E(z) \geq 0$ of our BM for all $z \in \text{CONF}$. Denote $E^* = \text{minimize}_{z \in \text{CONF}} E(z)$. Since $f \mapsto z = \text{range}(f)$ is a bijection from MAP to CONF , we must have

$$E^* = \text{minimize}_{z \in \text{CONF}} E(z) = \text{minimize}_{f \in \text{MAP}} E(\text{range}(f)).$$

Our goal is to minimize $\text{cost}(f)$, and we know that BM simulations should roughly minimize $E(z)$ over all $z \in \text{CONF}$. So, we define the BM energy function $E(z)$ by forcing

$$(10) \quad \text{cost}(f) = E(\text{range}(f))$$

for any registration mapping $f \in \text{MAP}$, which—due to the preceding subsection—is equivalent to

$$(11) \quad E(z) = \text{cost}(\text{map}(z)) \text{ for all configurations } z \in \text{CONF}.$$

The next subsection will explicitly express the energy $E(z)$ in terms of *cliques* of neurons. Due to (10) and (11) we have

$$E^* = \text{minimize}_{f \in \text{MAP}} \text{cost}(f) = \text{minimize}_{z \in \text{CONF}} E(z).$$

For large time t , the BM stochastic configuration $Z(t)$ tends with high probability to concentrate on configurations $z \in \text{CONF}$, which roughly minimize $E(z)$. The random registration $F^t = \text{map}(Z(t))$ will belong to MAP and verify $Z(t) = \text{range}(F^t)$, so that $E(Z(t)) = E(\text{range}(F^t)) = \text{cost}(F^t)$. Consequently, for large t —with high probability—the random mapping $F^t = \text{map}(Z(t))$ will have a value of the cost functional $\text{cost}(F^t)$ close to $\text{minimize}_{f \in \text{MAP}} \text{cost}(f)$.

12.3. Cliques of Interactive Neurons. The BM energy function $E(z)$ just defined turns out to involve only three sets of small cliques: **i)** CL_1 is the set of all singletons $K = \{i\}$, with $i = 1 \dots N$. **ii)** CL_2 is the set of all pairs $K = \{i, j\}$ such that cells b_i and b_j are *neighbors* in B . **iii)** CL_3 is the set of all triplets $K = \{i, j, k\}$ such that cells b_j and b_k are both *neighbors* of b_i in B . Denote $CLQ = CL_1 \cup CL_2 \cup CL_3$ the set of all cliques for our BM.

Cliques in CL_1 . For each clique $K = \{i\}$ in CL_1 , and each $z \in \text{CONF}$, define its energy $J_{\text{match}_K}(z) = J_{\text{match}_K}(z_i)$ by

$$J_{\text{match}_K}(z) = -\frac{1}{N} \log \text{LIK}(b_i, z_i) \text{ for all } z \in \text{ZW},$$

where LIK is given by (9). Set $J_{\text{match}_K} \equiv 0$ for K in $CL_2 \cup CL_3$. For all $z \in \text{CONF}$, define the energy $E_{\text{match}}(z)$ by

$$E_{\text{match}}(z) = \sum_{K \in CLQ} J_{\text{match}_K}(z) = \sum_{K \in CL_1} J_{\text{match}_K}(z),$$

which implies that the registration $f = \text{map}(z)$ verifies

$$\text{match}(f) = E_{\text{match}}(z).$$

Cliques in CL_2 . For all $z \in CONF$, all cliques $K = \{i, j\}$ in CL_2 , define the clique energies $J_{\text{over}_K}(z) = J_{\text{over}_K}(z_i, z_j)$ and $J_{\text{stab}_K}(z) = J_{\text{stab}_K}(z_i, z_j)$ by $J_{\text{over}_K}(z) = 1_{z_i=z_j}/N$ and

$$J_{\text{stab}_K}(z) = \frac{1}{N|G_i||G_j|} 1_{b_j \sim b_i} 1_{z_j \not\sim z_i},$$

where $|G_i|$ and $|G_j|$ are the numbers of neighbors in B for cells z_i and z_j , respectively. Set $J_{\text{over}_K} = J_{\text{stab}_K} \equiv 0$ for K in $CL_1 \cup CL_3$. Define the two energy functions

$$\begin{aligned} E_{\text{over}}(z) &= \sum_{K \in CL_Q} J_{\text{over}_K}(z) = \sum_{K \in CL_2} J_{\text{over}_K}(z), \\ E_{\text{stab}}(z) &= \sum_{K \in CL_Q} J_{\text{stab}_K}(z) = \sum_{K \in CL_2} J_{\text{stab}_K}(z), \end{aligned}$$

which implies that $f = \text{map}(z)$ verifies $\text{over}(f) = E_{\text{over}}(z)$ and $\text{stab}(f) = E_{\text{stab}}(z)$.

Cliques in CL_3 . For each clique $K = \{i, j, k\}$ in CL_3 , define the clique energy J_{flip_K} by

$$J_{\text{flip}_K}(z) = J_{\text{flip}_{i,j,k}}(z) = \frac{1}{N|G_i|^2} \text{FLIP}(f^{i,j,k}, b_i, b_j, b_k),$$

where $f^{i,j,k}$ is any registration mapping b_i, b_j, b_k onto z_i, z_j, z_k . The indicator FLIP was defined in Sec. 11.2. Set $J_{\text{flip}_K} \equiv 0$ for K in $CL_1 \cup CL_2$. Define the energy

$$E_{\text{flip}}(z) = \sum_{K \in CL_Q} J_{\text{flip}_K}(z) = \sum_{K \in CL_3} J_{\text{flip}_K}(z),$$

which implies that $f = F(z)$ verifies $\text{flip}(f) = E_{\text{flip}}(z)$.

Finally, define the clique energy J_K for all $K \in CL_Q$ by the linear combination

$$J_K = \lambda_{\text{match}} J_{\text{match}_K} + \lambda_{\text{over}} J_{\text{over}_K} + \lambda_{\text{stab}} J_{\text{stab}_K} + \lambda_{\text{flip}} J_{\text{flip}_K}.$$

Summing this relation over all $K \in CL_Q$ yields

$$(12) \quad \sum_{K \in CL_Q} J_K = \lambda_{\text{match}} E_{\text{match}} + \lambda_{\text{over}} E_{\text{over}} + \lambda_{\text{stab}} E_{\text{stab}} + \lambda_{\text{flip}} E_{\text{flip}}.$$

Define then the final BM energy function $z \mapsto E(z)$ by

$$(13) \quad E(z) = \sum_{K \in CL_Q} J_K(z) \text{ for all } z \text{ in } CONF.$$

For any $z \in CONF$, the associated registration $f = \text{map}(z)$ verifies $\text{match}(f) = E_{\text{match}}(z)$, $\text{over}(f) = E_{\text{over}}(z)$, $\text{stab}(f) = E_{\text{stab}}(z)$, $\text{flip}(f) = E_{\text{flip}}(z)$. By weighted linear combination of these equalities, and due to (12), we obtain for all configurations $z \in CONF$, $E(z) = \text{cost}(f)$ when $f = \text{map}(z)$ or, equivalently, when $z = \text{range}(f)$.

12.4. Test Set of 100 Synthetic Image Pairs. As shown above, the minimization of $\text{cost}(f)$ over all registrations $f \in MAP$ is equivalent to seeking BM configurations $z \in CONF$ with minimal energy $E(z)$. We have implemented this minimization of $E(z)$ by the long term asynchronous dynamics of the BM just defined. This algorithm was designed for the registration of image pairs exhibiting no cell division, and was, therefore, implemented after the automatic reduction of the generic registration problem, as indicated earlier. We have tested this specialized registration algorithm on a set $BENCH_{100}$ of 100 pairs of successive images of simulated cell colonies exhibiting no cell divisions. These 100 image pairs were extracted from the benchmark set $BENCH_6$ of synthetic image sequence described in section Sec. 2. The 100 pairs of cell sets B, B_+ had sizes $N = \text{card}(B) = \text{card}(B_+)$ ranging from 80 to 100 cells.

For each test pair B, B_+ , each target window $W(j)$ typically contained 30 to 40 cells. The set $CONF$ of configurations had huge cardinality ranging from 10^{130} to 10^{160} . But the average number of neighbors of a cell was around 4 to 5.

12.5. Implementation of BM minimization for $\text{cost}(f)$. The numbers clq_1, clq_2, clq_3 of cliques in CL_1, CL_2, CL_3 have the following rough ranges $80 \leq clq_1 \leq 100$, $160 \leq clq_2 \leq 250$, and $450 \leq clq_3 \leq 600$. For $k = 1, 2, 3$, denote $val(k)$ the numbers of non-zero values for $J_K(z)$ when z runs through $CONF$ and K runs through all cliques of cardinality k . One easily checks the rough upper bounds $val(1) < 4.00e3$; $val(2) < 2.00e5$; $val(3) < 3.00e5$. Hence, to automatically register B to B_+ , one could pre-compute and store all the possible values of $J_K(z)$ for all cliques $K \in CL_1 \cup CL_2 \cup CL_3$ and all the configurations $z \in CONF$. This accelerates the key computing steps of the asynchronous BM dynamics, namely, for the evaluation of energy change $\Delta E = E(z') - E(z)$, when configurations z and z' differs at only one site $j \in S$. Indeed, the single site modification $z_j \rightarrow z'_j$ affects only the energy values $J_K(z)$ for the very small number $r(j)$ of cliques K , which contain the site j . In our benchmark sets of synthetic images, one had $r(j) < 24$ for all $j \in S$. Hence, the computation of ΔE was fast since it requires retrieving at most 24 pairs of pre-computed $J_K(z)$, $J_K(z')$, and evaluating the 24 differences $J_K(z') - J_K(z)$. Another practical acceleration step is to replace the ubiquitous computations of probabilities $p(t) = \exp(-D/Temp(t))$ by simply testing the value $-D/Temp(t)$ against 100 precomputed logarithmic thresholds.

In our implementation of ABM dynamics, we used virtual temperature schemes such as $Temp(t) = 50 \cdot \rho^t$ with $0.995 \leq \rho \leq 0.999$. The BM simulation was stopped when the stochastic energy $E(Z(t))$ had remained roughly stable during the last N steps. Since all target windows $W(j)$ had cardinality smaller 40, the initial configuration $Z(0) = x$ was computed via

$$x_j = \underset{y \in W(j)}{\operatorname{argmax}} \operatorname{LIK}(b_j, y) \quad \text{for } j = 1, \dots, N,$$

where the likelihoods LIK were defined by (9).

12.6. Weight Calibration. For the pair of successive *synthetic* images J, J_+ displayed in Fig. 4, we have $N = \operatorname{card}(B) = \operatorname{card}(B_+) = 513$ cells. The ground truth registration f is known by construction; we used it to apply the weight calibration described in Sec. 5. We set the meta-parameter γ to $1e10$ and obtained the vector of weights

$$(14) \quad \Lambda^* = [\lambda_{\text{match}}^*, \lambda_{\text{over}}^*, \lambda_{\text{stab}}^*, \lambda_{\text{flip}}^*] = [110, 300, 300, 290].$$

These weights are *kept fixed for all the 100 pairs* of images in the set BENCH_{100} . The determined weights are used in the cost function $\text{cost}(f)$ defined above. This correctly parametrized the BM energy function $E(z)$. We then simulated the BM stochastic dynamics to minimize the BM energy $E(Z(t))$.

12.7. BM Simulations. We launched 100 simulations of the asynchronous BM dynamics, one for each pair of successive images in our test set BENCH_{100} . For each such pair, the ground truth mapping $f : B \rightarrow B_+$ was known by construction and the stochastic minimization of the BM energy generated an estimated cells registration $f' : B \rightarrow B_+$. For each pair B, B_+ in BENCH_{100} , the accuracy of this automatically computed registration f' was evaluated by the percentage of cells $b \in B$ such that $f'(b) = f(b)$. When $\operatorname{card}(B) = N$, our BM has N stochastic neurons, and the asynchronous BM dynamics proceeds by successive *epochs*. Each epoch is a sequence of N single site updates of the BM configuration. For each one of our 100 simulations of BM asynchronous dynamics, the number of epochs ranged from 250 to 450.

The average computing time was about eight seconds per epoch on a standard laptop, which entailed a computing time ranging from 30 to 50 minutes for each one of our 100 automatic registrations $f' : B \rightarrow B_+$. Note that the temperature scheme had not been optimized yet, so that these computing times are upper bounds. Earlier SBM studies [13, 15] indicate that the same energy minimizations on GPUs could provide a computational speedup by a factor ranging between 30 and 50.

We report registration accuracies in Tab. 2. For each pair of images in BENCH_{100} , the accuracy of automatic registration was larger than 94.5%. The overall average registration accuracy was quite high at 99%.

13. Registration for Cell Dynamics Involving Growth, Motion, and Cell Divisions.

13.1. Tests of Cell Registration algorithms on Synthetic Data. We now consider more generic long synthetic image sequences of simulated cell colonies, with a small interframe duration of one minute. We still impose the mild constraint that no cell is lost between two successive images. The main difference with the earlier benchmark BENCH_{100} is that cells are *allowed to freely divide* during interframes, as well as to grow and to move. For the full implementation on 100 pairs of successive images, we first execute the parent-children pairing, and remove the identified parent-children triplets; we can then apply our cell registration algorithmic on the reduced sets cells. Our image sequence contained 760 true parent-children triplets, which we automatically identified with an accuracy of 100%. As outlined earlier, we removed all these identified cell triplets and then applied our tracking algorithm. This

TABLE 2

Registration accuracy for synthetic image sequence $BENCH_{100}$. We consider 100 pairs of consecutive synthetic images (image sequence $BENCH_{100}$). Automatic registration was implemented by BM minimization of the cost function $\text{cost}(f)$, which was parametrized by the vector of optimized weights Λ^* in (14). The average registration accuracy was 99%.

registration accuracy	number of frames
$\text{acc} = 100\%$	55 frames out of 100
$99\% \geq \text{acc} > 97\%$	40 frames out of 100
$96\% \geq \text{acc} > 94.5\%$	5 frames out of 100

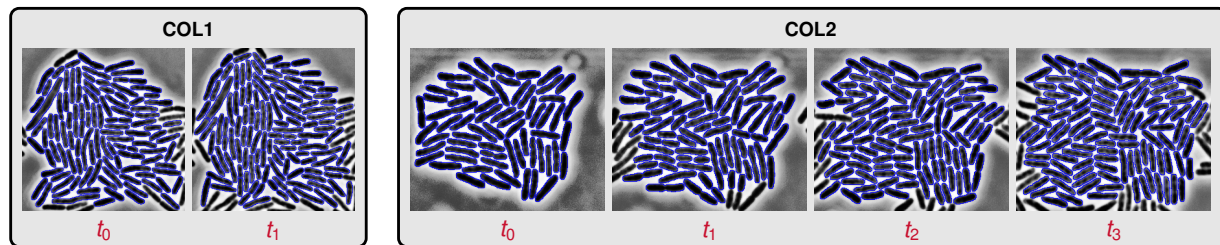


FIG. 6. **Segmentation results for experimental recordings of live cell colonies.** We show two short image sequences extracts COL1 (left) and COL2 (right). The interframe duration is six minutes. The image sequence extract COL1 has only two successive image frames. The image sequence extract COL2 has four successive image frames. We are going to automatically compute four cell registrations, one for each pair of successive images in COL1 and COL2.

left us with a total of 1.26×10^4 cells (spread over 100 frames). Full automatic registration was then implemented with an accuracy higher than 99.5%.

13.2. Tests of Cell Registration algorithms on Laboratory Image Sequences. To test our cell tracking algorithm on pairs of consecutive images extracted from recorded image sequences of bacterial colonies, we had to automatically delineate all individual cells in each image. We use the Watershed algorithm [31] (also used, e.g., in [53]) to segment each frame into individual image segments containing one single cell each. In a second step, we then identified the contour of each single cell b by applying the Mumford-Shah algorithm [37, 68] within the image segment containing a cell b .

We then apply ad hoc nonlinear filters to remove minor segmentation artifacts. Since this procedure is quite time consuming for large images, we have implemented it to produce a training set of delineated individual cells to train a CNN for image segmentation. After automatic training, this CNN substantially reduces the runtime of the cell segmentation/delineation procedure. We show the resulting segmentations in Fig. 6.

After each cell has been identified (i.e., segmented out) in each pair J, J_+ of successive images, we transform J, J_+ into binary images, where cells appear in white on a black background. For each resulting pair B, B_+ of successive sets of cells, we apply the parent-children pairing algorithm outlined in Sec. 6 to identify all the short lineages. For the two successive images in COL1, the discovered short lineages are shown in Fig. 7 (left pair of images). Here, color designates the cell triplet algorithmically identified: parent cell in image J and its two children in image J_+ . We then remove each identified “parent” from B and its two children from B_+ . This yields the reduced cell sets $redB$ and $redB_+$. We can then apply our tracking algorithm (see 10) dedicated to situations where cells do not divide during the interframe.

For image sequences of live cell colonies we had to re-calibrate most of our weight parameters. The weight parameters used for these image sequences are summarized in Tab. 3.

The BM temperature scheme was $Temp(t) = 2000(0.995)^t$, with the number of epochs capped at 5000. We illustrate our COL1 automatic registration results in Fig. 7 (right pair of images). Here, if cell $b \in redB$ has been automatically registered onto cell $b_+ \in redB_+$, b, b_+ share the same color. The cells colored in white in $redB_+$ are cells which the registration algorithm did not succeed in matching to some cell in $redB$. These errors can essentially be attributed to errors in the parent-children pairing step. By visual inspection we have determined that there are 14 true parent-children triplets in the successive images of COL1. Our parent-children pairing algorithm did correctly identify 11 of these 14 triplets. To check further the performance of our registration algorithm on live images, we also report automatic registration results for “manually prepared” true versions of $redB$ and $redB_+$, obtained by removing “manually” the true parent-children triplets determined by visual inspection. For the short image sequence COL2, results

TABLE 3
Cost function weights for parent-children pairing in the COLI images displayed in Fig. 6.

Weights	λ_{cen}	λ_{siz}	λ_{ang}	λ_{gap}	λ_{dev}	λ_{rat}	λ_{rank}	λ_{over}
Value	3	7	100	.8	4	.01	.01	600

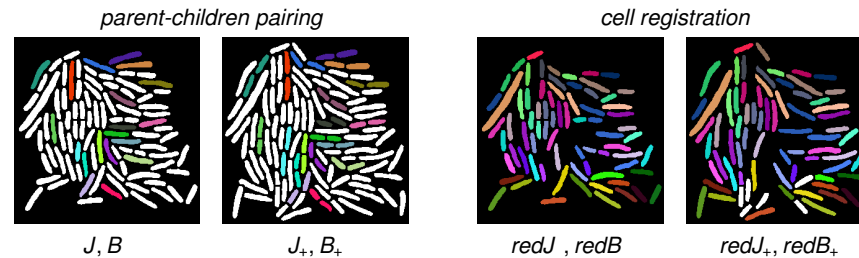


FIG. 7. Cell tracking results for the pair COLI of successive images J, J_+ shown in Fig. 6. The interframe duration is six minutes. Left: Results for parent-children pairing on COLI. Automatically detected parent-children triplets are displayed in the same color. Right: Computed registration. The removal of the automatically detected parent-children triplets (see left column) generates the reduced cell sets redB and redB_+ . Automatic registration of redB and redB_+ is again displayed via identical color for the registered cell pairs (b, b_+) . Mismatches are mostly due to previous errors in parent-children pairing (see Fig. 8 for a more detailed assessment).

are displayed in Fig. 8.

The display setup is the same: The left column shows the results of automatic parent-children pairing. The middle column illustrates the computed registration after automatic removal of the computer identified parent-children triplets. The third column displays the computed registration after removing “manually” the true parent-children triplets determined by visual inspection. Note that the overall matching accuracy can be improved if we reduce errors in the parent-children pairing. We report quantitative accuracies in Tab. 4. For parent-children pairing, accuracy ranges between 70% and 78%. For pure registration after correct parent-children pairing, accuracy ranges between 90% and 100%.

14. Conclusions and Future Work. We have developed a methodology for automatic cell tracking in recordings of dense bacterial colonies growing in a mono-layer. We have also validated our approach using synthetic data from agent based simulations, as well as experimental recordings of *E. coli* colonies growing in microfluidic traps. Our next goal is to streamline our implementation for systematic cell registration on experimentally acquired recordings of such cell colonies, to enable automated quantitative analysis and modeling of cell population dynamics and lineages.

There are a number of challenges for our cell tracking algorithm: Inherent imaging artifacts such as noise or intensity drifts, cells overlaps, similarity of cell shape characteristics across the population, tight packing of cells, somewhat large interframe times, cell growth combined with cell motion and cell divisions, represent just a few of these challenges. Overall, the cell tracking problem has combinatorial complexity, and for large frames is beyond the concrete patience of human experts. We tackle these challenges by developing a two-stage algorithm that first identifies parent-children triplets and subsequently computes cell registration from one frame to the next, after reducing the two original cell sets by automatic removal of the identified parent-children triplets. Our algorithms specify innovative cost functions dedicated to these registration challenges. These cost functions have combinatorial complexity. To discover good registrations we minimize these cost functions numerically by intensive stochastic simulations of specifically structured BMs. We have validated the potential of our approach by reporting promising results obtained on long synthetic image sequences of simulated cell colonies (which naturally provide a ground truth for cell registration from one frame to the next). We have also successfully tested our algorithms on experimental recordings of live bacterial colonies.

In future work we will further improve the stability and accuracy of our cell registration algorithms by exploring natural modifications of our cost functions, in particular to improve the accuracies of our automatic detection of cell divisions, and to handle also the unavoidable new cell arrivals into or departures from the current field of vision. Moreover, we will work on our BM simulation algorithms to deploy their stochastic dynamics in semi synchronous formats on parallel computing platforms (in particular, graphic processing units) to drastically reduce their runtime.

Acknowledgements. This work was partly supported by the National Science Foundation (NSF) through the grants DMS-1854853 (AM & RA), DMS-2009923 (AM & RA), 1662305 (KJ), MCB-1936770 (KJ), and DMS-

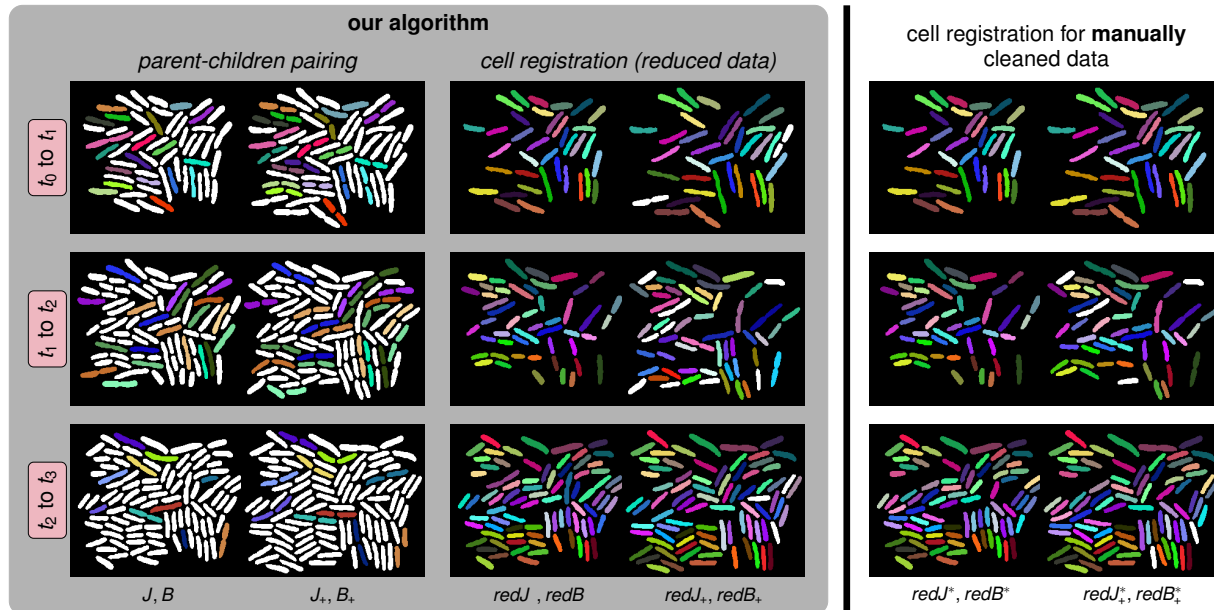


FIG. 8. **Cell tracking results for the short image sequence COL2 in Fig. 6.** The interframe duration for COL2 is six minutes. COL2 involves four successive images $J(t_i)$, $i = 0, 1, 2, 3$. In our figure, each one of the three rows displays the automatic cell registration results between images $J(t_i)$ and $J(t_{i+1})$ for $i = 0, 1, 2$. We report the accuracies of parent-children pairing and of the registration in Tab. 4. Left column: Results for parent-children pairing. Each parent-children triplet is identified by the same color for each parent cell and its two children. Middle column: Display of the automatically computed registration after removing the parent-children triplets already identified in order to generate two reduced sets $redB$ and $redB_+$ of cells. Again, the same color is used for each pair of automatically registered cells. The white cells in $redB_+$ are cells which could not be registered to some cell in $redB$. Right column: To differentiate between errors induced during automatic identification of and errors generated by automatic registration between $redB$ and $redB_+$, we manually removed all “true” parent-children triplets and then applied our registration algorithm to this “cleaned” (reduced) cell sets $redB^*$ and $redB_+^*$.

TABLE 4

Cell tracking accuracy for the short image sequence COL2 in Fig. 6 with an interframe of six minutes. We report the ratio of correctly predicted cell matches over the total number of true cell matches and the associated percentages. The accuracy results quantify four distinct percentages of correct detections (i) for parent cells in image J , (ii) for children cells in image J_+ , (iii) for parent-children triplets, and (iv) for registered pairs of cells $(b, b_+) \in redB \times redB_+$.

task	accuracy					
		$\{t_0, t_1\}$		$\{t_1, t_2\}$		$\{t_2, t_3\}$
correctly detected parents	15/19	79%	20/21	95%	7/10	70%
correctly detected children	35/38	92%	32/42	76%	14/20	70%
correct parent-children triplets	15/19	78%	16/21	76%	7/10	70%
correctly registered cell pairs	36/36	100%	44/49	90%	76/80	95%

2012825 (AM); the joint NSF-National Institutes of General Medical Sciences Mathematical Biology Program grant DMS-1662290 (MRB); and the Welch Foundation grant C-1729 (MRB). Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the NSF or the Welch Foundation. This work was completed in part with resources provided by the Research Computing Data Core at the University of Houston.

Appendix A. Stochastic Dynamics of BMs.

Notations and terminology refer to Sec. 7. Consider a BM network of N stochastic neurons U_j , with finite configuration set $CONF = W(1) \times \dots \times W(N)$. At time t , let $Z_j(t) \in W(j)$ be the random state of neuron U_j , and the BM configuration $Z(t) \in CONF$ is then $Z(t) = \{Z_1(t), \dots, Z_N(t)\}$. Fix as in Sec. 7 a sequence $Temp(t)$ of virtual temperatures slowly decreasing to 0 for large t .

There are two main options to implement the Markov chain dynamics $Z(t) \rightarrow Z(t+1)$ (see [10]).

Asynchronous BM Dynamics. Generate a long random sequence of sites $m(t) \in S = \{1, \dots, N\}$, for instance by concatenating successive random permutations of the set S . At time t , the only neuron which may modify its current

state is $U_{m(t)}$. For brevity, write $M = m(t)$. The neuron U_M will compute its new random state $Z_M(t+1) \in W(M)$ by the following *updating procedure*:

- For each y in $W(M)$, define a new configuration $Y \in \text{CONF}$ by $Y_M(t) = y$, and $Y_j(t) = Z_j(t)$ for all $j \neq M$. Let $\Delta(y) = E(Y) - E(Z(t))$ be the corresponding BM energy change.
- In the finite set $W(M)$, select any z such that $\Delta(z) = \min_{y \in W(M)} \Delta(y)$, and set $D = \max\{0, \Delta(z)\}$.
- Compute the probability $p = \exp(-D/\text{Temp}(t))$.
- The new random state $Z_M(t+1)$ of neuron U_M will be equal to z with probability p and equal to the current state $Z_M(t)$ with probability $1 - p$.
- For all $j \neq M$, the new state $Z_j(t+1)$ of neuron U_j remains equal to its current state $U_j(t)$.

Synchronous BM Dynamics. Fix a *synchrony* parameter $0 < \alpha < 1$, usually around 50%. At each time t , all neurons U_j synchronously, but independently compute their own random *binary tag* $\text{tag}_j(t)$, equal to 1 with probability α , and to 0 with probability $(1 - \alpha)$. Let $\text{SYN}(t)$ be the set of all neurons. All the neurons U_j such that $\text{tag}_j(t) = 1$ then synchronously and independently compute their new random states $Z_j(t+1) \in W(j)$ by applying the updating procedure given above. And for all j such that $\text{tag}_j(t) = 0$, the new state $Z_j(t+1)$ of U_j remains equal to $Z_j(t)$.

Comparing Asynchronous and Synchronous BM Dynamics. As t becomes large, and for temperatures $\text{Temp}(t)$ slowly decreasing to 0, both BM dynamics generate with high probability configurations $Z(t)$ which provide deep local minima $E(Z(t))$ of the BM energy function. The asynchronous dynamics can be fairly slow. But the synchronous dynamics is much faster since it emulates efficient forms of *parallel simulated annealing* (see [11, 75]) and is directly implementable on GPUs.

REFERENCES

- [1] D. H. ACKLEY, G. E. HINTON, AND T. J. SEJNOWSKI, *A learning algorithm for boltzmann machines*, Cognitive Science, 9 (1985), pp. 147–169.
- [2] A. AGRAWAL, R. VERSCHUEREN, S. DIAMOND, AND S. BOYD, *A rewriting system for convex optimization problems*, Journal of Control and Decision, 5 (2018), pp. 42–60.
- [3] S. U. AKRAM, J. KANNALA, L. EKLUND, AND J. HEIKKILÄ, *Joint cell segmentation and tracking using cell proposals*, in IEEE 13th International Symposium on Biomedical Imaging (ISBI), 2016, pp. 920–924.
- [4] Y. AL-KOFAHI, A. ZALTSMAN, R. GRAVES, W. MARSHALL, AND M. RUSU, *A deep learning-based algorithm for 2-d cell segmentation in microscopy images*, BMC Bioinformatics, 19 (2018), pp. 1–11.
- [5] R. N. ALNAHHAS, M. SADEGHPOUR, Y. CHEN, A. A. FREY, W. OTT, K. JOSIĆ, AND M. R. BENNETT, *Majority sensing in synthetic microbial consortia*, Nature Communications, 11 (2020), pp. 1–10.
- [6] R. N. ALNAHHAS, J. J. WINKLE, A. J. HIRNING, B. KARAMECHED, W. OTT, K. JOSIĆ, AND M. R. BENNETT, *Spatiotemporal Dynamics of Synthetic Microbial Consortia in Microfluidic Devices*, ACS Synthetic Biology, 8 (2019), pp. 2051–2058.
- [7] B. APPLETON AND H. TALBOT, *Globally optimal geodesic active contours*, Journal of Mathematical Imaging and Vision, 23 (2005), pp. 67–86.
- [8] A. ARBELLE, J. REYES, J.-Y. CHEN, G. LAHAV, AND T. R. RAVIV, *A probabilistic approach to joint cell tracking and segmentation in high-throughput microscopy videos*, Medical Image Analysis, 47 (2018), pp. 140–152.
- [9] R. AZENCOTT, *Synchronous boltzmann machines and artificial vision*, Neural Networks, (1990), pp. 135–143.
- [10] R. AZENCOTT, *Synchronous boltzmann machines and Gibbs fields: Learning algorithms*, in Neurocomputing, Springer, 1990, pp. 51–63.
- [11] R. AZENCOTT, *Simulated annealing: Parallelization techniques*, vol. 27, Wiley-Interscience, 1992.
- [12] R. AZENCOTT, B. CHALMOND, AND F. COLDEFY, *Markov image fusion to detect intensity valleys*, International Journal of Computer Vision, 16 (1994), pp. 135–145.
- [13] R. AZENCOTT, A. DOUTRIAUX, AND L. YOUNES, *Synchronous Boltzmann Machines and curve identification tasks*, Network: Computation in Neural Systems, 4 (1993), pp. 461–480.
- [14] R. AZENCOTT, C. GRAFFIGNE, AND C. LABOURDETTE, *Edge detection and textured images segmentation*, in Stochastic Models in Image Analysis, vol. 74, 1992, pp. 75–88.
- [15] R. AZENCOTT AND J. LACAILLE, *Smooth image contours and boltzmann machines*, in Proc. of the International Conference on Neural Nets, 1991.
- [16] A. D. BALOMENOS, P. TSAKANIKAS, Z. ASPRIDOU, A. P. TAMPAKAKI, K. P. KOUTSOUMANIS, AND E. S. MANOLAKOS, *Image analysis driven single-cell analytics for systems microbiology*, BMC Systems Biology, 11 (2017), pp. 1–21.
- [17] A. D. BALOMENOS, P. TSAKANIKAS, AND E. S. MANOLAKOS, *Tracking single-cells in overcrowded bacterial colonies*, in Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2015, pp. 6473–6476.
- [18] D. S. BANERJEE, G. STEPHENSON, AND S. G. DAS, *Segmentation and analysis of mother machine data: Sam*, bioRxiv, (2020).
- [19] M. R. BENNETT AND J. HASTY, *Microfluidic devices for measuring gene network dynamics in single cells*, Nature Reviews Genetics, 10 (2009), pp. 628–638.
- [20] R. BISE, K. LI, S. EOM, AND T. KANADE, *Reliably tracking partially overlapping neural stem cells in DIC microscopy image sequences*, in Proc International Conference on Medical Image Computing and Computer-Assisted Intervention Workshop, 2009, pp. 67–77.
- [21] R. BISE AND Y. SATO, *Cell detection from redundant candidate regions under non-overlapping constraints*, IEEE Transactions on Medical Imaging, 34 (2015), pp. 1417–1427.
- [22] R. BISE, Z. YIN, AND T. KANADE, *Reliable cell tracking by global data association*, in IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011, pp. 1004–1010.

- [23] A. BORZI, K. ITO, AND K. KUNISCH, *An optimal control approach to optical flow computation*, International Journal for Numerical Methods in Fluids, 40 (2002), pp. 231–240.
- [24] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, 2004.
- [25] C. J. BUTTS-WILMSMEYER, S. RAPP, AND B. GUTHRIE, *The technological advancements that enabled the age of big data in the environmental sciences: A history and future directions*, 2020.
- [26] A. E. CARPENTER, T. R. JONES, M. R. LAMPRECHT, C. CLARKE, I. H. KANG, O. FRIMAN, D. A. GUERTIN, J. H. CHANG, R. A. LINDQUIST, J. MOFFAT, P. GOLLAND, AND D. M. SABATINI, *CellProfiler: Image analysis software for identifying and quantifying cell phenotypes*, Genome Biology, 7 (2006), p. R100.
- [27] A. CHAKRABORTY AND A. K. ROY-CHOWDHURY, *Context aware spatio-temporal cell tracking in densely packed multilayer tissues*, Medical Image Analysis, 19 (2015), pp. 149–163.
- [28] Y. CHEN, J. K. KIM, A. J. HIRNING, K. JOSIĆ, AND M. R. BENNETT, *Emergent genetic oscillations in a synthetic microbial consortium*, Science, 349 (2015), pp. 986–989.
- [29] T. DANINO, O. MONDRAGÓN-PALOMINO, L. TSMIRING, AND J. HASTY, *A synchronized quorum of genetic clocks*, Nature, 463 (2010), pp. 326–330, <https://arxiv.org/abs/15334406>.
- [30] S. DIAMOND AND S. BOYD, *CVXPY: A Python-embedded modeling language for convex optimization*, Journal of Machine Learning Research, 17 (2016), pp. 1–5.
- [31] H. DIGABEL AND C. LANTUEJOL, *Iterative algorithms*, in Proceedings of the 2nd European Symposium Quantitative Analysis of Microstructures in Material Science, Biology and Medicine, 1978, pp. 85–89.
- [32] Z. DLAMINI, F. Z. FRANCIS, R. HULL, AND R. MARIMA, *Artificial intelligence (AI) and big data in cancer and precision oncology*, 2020.
- [33] N. EL NAJJAR, M. C. VAN TEESELENG, B. MAYER, S. HERMANN, M. THANBICHLER, AND P. L. GRAUMANN, *Bacterial cell growth is arrested by violet and blue, but not yellow light excitation during fluorescence microscopy*, BMC Molecular and Cell Biology, 21 (2020), p. 35.
- [34] T. FALK, D. MAI, R. BENSCH, Ö. ÇIÇEK, A. ABDULKADIR, Y. MARRAKCHI, A. BÖHM, J. DEUBNER, Z. JÄCKEL, K. SEIWALD, ET AL., *U-Net: deep learning for cell counting, detection, and morphometry*, Nature Methods, 16 (2019), pp. 67–70.
- [35] A. J. HAND, T. SUN, D. C. BARBER, D. R. HOSE, AND S. MACNEIL, *Automated tracking of migrating cells in phase-contrast video microscopy sequences using image registration*, Journal of Microscopy, 234 (2009), pp. 62–79.
- [36] J. HAYASHIDA, K. NISHIMURA, AND R. BISE, *MPM: Joint representation of motion and position map for cell tracking*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3823–3832.
- [37] M. HINTERMÜLLER AND W. RING, *An inexact Newton-CG-type active contour approach for the minimization of the Mumford-Shah functional*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 19–42.
- [38] G. E. HINTON AND T. J. SEJNOWSKI, *Parallel distributed processing: explorations in the microstructure of cognition*, MIT Press, 1986, ch. Learning and relearning in Boltzmann machines, pp. 282–317.
- [39] B. K. P. HORN AND B. G. SHUNCK, *Determining optical flow*, Artificial Intelligence, 17 (1981), pp. 185–203.
- [40] J. ICHA, M. WEBER, J. C. WATERS, AND C. NORDEN, *Phototoxicity in live fluorescence microscopy, and how to avoid it*, BioEssays, 39 (2017), p. 1700003.
- [41] J. JEAN-PASCAL, M. DIMICCOLLI, AND L. MOISAN, *Active skeleton for bacteria modelling*, Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization, 5 (2017), pp. 274–286.
- [42] F. JUG, T. PIETZSCH, D. KAINMÜLLER, J. FUNKE, M. KAISER, E. VAN NIMWEGEN, C. ROTHER, AND G. MYERS, *Optimal joint segmentation and tracking of Escherichia coli in the mother machine*, in Bayesian and graphical Models for Biomedical Imaging, vol. LNCS 8677, 2014, pp. 25–36.
- [43] L. KAMENSKY, T. R. JONES, A. FRASER, M. BRAY, D. LOGAN, K. MADDEN, V. LJOSA, C. RUEDEN, G. B. HARRIS, K. ELICEIRI, AND A. E. CARPENTER, *Improved structure, function, and compatibility for cellprofiler: modular high-throughput image analysis software*, Bioinformatics, 27 (2011), pp. 1179–1180.
- [44] T. KANADE, Z. YIN, R. BISE, S. HUH, S. EOM, M. F. SANDBOTHE, AND M. CHEN, *Cell image analysis: Algorithms, system and applications*, in 2011 IEEE Workshop on Applications of Computer Vision (WACV), IEEE, 2011, pp. 374–381.
- [45] C. KERVRANN AND A. TRUBUIL, *Optimal level curves and global minimizers of cost functionals in image segmentation*, Journal of Mathematical Imaging and Vision, 17 (2002), pp. 153–174.
- [46] J. K. KIM, Y. CHEN, A. J. HIRNING, R. N. ALNAHHAS, K. JOSIĆ, AND M. R. BENNETT, *Long-range spatio-temporal coordination of gene expression in synthetic microbial consortia*, Nature Chemical Biology, 15 (2019), pp. 1102–1109.
- [47] C. KIRISITS, L. F. LANG, AND O. SCHERZER, *Optical flow on evolving surfaces with space and time regularisation*, Journal of Mathematical Imaging and Vision, 52 (2015), pp. 55–70.
- [48] J. KLEIN, S. LEUPOLD, I. BIEGLER, R. BIEDENDIECK, R. MÜNCH, AND D. JAHN, *TLM-Tracker: Software for cell segmentation, tracking and lineage analysis in time-lapse microscopy movies*, Bioinformatics, 28 (2012), pp. 2276–2277.
- [49] A. KONG AND R. AZENCOTT, *Binary markov random fields and interpretable mass spectra discrimination*, Statistical applications in genetics and molecular biology, 16 (2017), pp. 13–30.
- [50] T. KROEGER, R. TIMOFTE, D. DAI, AND L. VAN GOOL, *Fast optical flow using dense inverse search*, in European Conference on Computer Vision, Springer, 2016, pp. 471–488.
- [51] E. LEE AND M. GUNZBURGER, *An optimal control formulation of an image registration problem*, Journal of Mathematical Imaging and Vision, 36 (2010), pp. 69–80.
- [52] K. LI, E. D. MILLER, M. CHEN, T. KANADE, L. E. WEISS, AND P. G. CAMPBELL, *Cell population tracking and lineage construction with spatiotemporal context*, Medical Image Analysis, 12 (2008), pp. 546–566.
- [53] M. LIU, A. CHAKRABORTY, D. SINGH, R. K. YADAV, G. MEENAKSHISUNDARAM, G. V. REDDY, AND A. ROY-CHOWDHURY, *Adaptive cell segmentation and tracking for volumetric confocal microscopy images of a developing plant meristem*, Molecular Plant, 4 (2011), pp. 922–931.
- [54] M. LIU, R. K. YADAV, A. ROY-CHOWDHURY, AND G. V. REDDY, *Automated tracking of stem cell lineages of Arabidopsis shoot apex using local graph matching*, The Plant Journal, 62 (2010), pp. 135–147.
- [55] J. C. W. LOCKE AND M. B. ELOWITZ, *Using movies to analyse gene circuit dynamics in single cells*, Nature Reviews Microbiology, 7 (2009), pp. 383–392.

- [56] K. LÖFFLER, T. SCHERR, AND R. MIKUT, *A graph-based cell tracking algorithm with few manually tunable parameters and automated segmentation error correction*, bioRxiv, (2021).
- [57] B. D. LUCAS AND T. KANADE, *An iterative image registration technique with an application to stereo vision*, in Proceedings of the International Conference on Artificial Intelligence, 1981, pp. 674–679.
- [58] J.-B. LUGAGNE, H. LIN, AND M. J. DUNLOP, *DeLTA: Automated cell segmentation, tracking, and lineage reconstruction using deep learning*, PLoS Computational Biology, 16 (2020), p. e1007673.
- [59] F. LUX AND P. MATULA, *DIC image segmentation of dense cell populations by combining deep learning and watershed*, in 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), 2019, pp. 236–239.
- [60] A. MANG AND G. BIROS, *An inexact Newton–Krylov algorithm for constrained diffeomorphic image registration*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 1030–1069.
- [61] A. MANG, A. GHOLAMI, C. DAVATZIKOS, AND G. BIROS, *CLAIRE: A distributed-memory solver for constrained large deformation diffeomorphic image registration*, SIAM Journal on Scientific Computing, 41 (2019), pp. C548–C584.
- [62] A. MANG AND L. RUTHOTTO, *A Lagrangian Gauss–Newton–Krylov solver for mass- and intensity-preserving diffeomorphic image registration*, SIAM Journal on Scientific Computing, 39 (2017), pp. B860–B885.
- [63] S. M. MARVASTI-ZADEH, L. CHENG, H. GHANEI-YAKHDAN, AND S. KASAEI, *Deep learning for visual tracking: A comprehensive survey*, IEEE Transactions on Intelligent Transportation Systems, (2021).
- [64] M. MASKA, V. ULMAN, D. SVOBODA, P. MATULA, P. MATULA, C. EDERRA, A. URBIOLA, T. ESPANA, S. VENKATESAN, D. M. W. BALAK, P. KARAS, T. BOLCKOVA, M. STREITOVA, C. CARHEL, S. CORALUPPI, N. HARDER, K. ROHR, K. E. G. MAGNUSSON, J. JALDEN, H. M. BLAU, O. DZYUBACHYK, P. KRIZEK, G. M. HAGEN, D. PASTOR-ESCUREDO, D. JIMENEZ-CARRETERO, M. J. LEDESMA-CARBAYO, A. MUNOZ-BARRUTIA, E. MEIJERING, M. KOZUBEK, AND C. O. DE SOLORZANO, *A benchmark for comparison of cell tracking algorithms*, Bioinformatics, 30 (2014), pp. 1609–1617.
- [65] W. MATHER, O. MONDRAGON-PALOMINO, T. DANINO, J. HASTY, AND L. S. TSMIRING, *Streaming instability in growing cell populations*, Physical Review Letters, 104 (2010).
- [66] C. MCQUIN, A. GOODMAN, V. CHERNYSHEV, L. KAMENSKY, B. A. CIMINI, K. W. KARHOHS, M. DOAN, L. DING, S. M. RAFELSKI, D. THIRSTRUP, W. WIEGRAEBE, S. SINGH, T. BECKER, J. C. CAICEDO, AND A. E. CARPENTER, *CellProfiler 3.0: Next-generation image processing for biology*, PLoS Biology, 16 (2018), p. e2005970.
- [67] E. MOEN, D. BANNON, T. KUDO, W. GRAF, M. COVERT, AND D. VAN VALEN, *Deep learning for cellular image analysis*, Nature Methods, 16 (2019), pp. 1233–1246.
- [68] D. B. MUMFORD AND J. SHAH, *Optimal approximations by piecewise smooth functions and associated variational problems*, Communications on Pure and Applied Mathematics, (1989).
- [69] K. NISHIMURA, J. HAYASHIDA, C. WANG, AND R. BISE, *Weakly-supervised cell tracking via backward-and-forward propagation*, in Proc. European Conference on Computer Vision, 2019, pp. 104–121.
- [70] K. OKUMA, A. TALEGHANI, N. DE FREITAS, J. J. LITTLE, AND D. G. LOWE, *A boosted particle filter: Multitarget detection and tracking*, in European Conference on Computer Vision, Springer, 2004, pp. 28–39.
- [71] J. OLLION, M. ELEZ, AND L. ROBERT, *High-throughput detection and tracking of cells and intracellular spots in mother machine experiments*, Nature Protocols, 14 (2019), pp. 3144–3161.
- [72] C. PAYER, D. ŠTERN, M. FEINER, H. BISCHOF, AND M. URSCHLER, *Segmenting and tracking cell instances with cosine embeddings and recurrent hourglass networks*, Medical Image Analysis, 57 (2019), pp. 106–119.
- [73] C. PAYER, D. ŠTERN, T. NEFF, H. BISCHOF, AND M. URSCHLER, *Instance segmentation and tracking with cosine embeddings and recurrent hourglass networks*, in Proc Medical Image Computing and Computer Assisted Intervention, no. LNCS 11071, 2018, pp. 3–11.
- [74] M. PRIMET, A. DEMAREZ, F. TADDEI, A. LINDNER, AND L. MOISAN, *Tracking of cells in a sequence of images using a low-dimensional image representation*, in Proc IEEE International Symposium on Biomedical Imaging, 2008, pp. 995–998.
- [75] D. J. RAM, T. SREENIVAS, AND K. G. SUBRAMANIAM, *Parallel simulated annealing algorithms*, Journal of parallel and distributed computing, 37 (1996), pp. 207–212.
- [76] M. REMPLER, S. KUMAR, V. STIERLE, P. PAULITSCHKE, B. ANDRES, AND B. H. MENZE, *Cell lineage tracing in lens-free microscopy videos*, in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2017, pp. 3–11.
- [77] M. REMPLER, V. STIERLE, K. DITZEL, S. KUMAR, P. PAULITSCHKE, B. ANDRES, AND B. H. MENZE, *Tracing cell lineages in videos of lens-free microscopy*, Medical Image Analysis, 48 (2018), pp. 147–161.
- [78] O. RONNEBERGER, P. FISCHER, AND T. BROX, *U-Net: Convolutional networks for biomedical image segmentation*, in Proc Medical Image Computing and Computer Assisted Intervention, vol. LNCS 9351, 2015, pp. 234–241.
- [79] J. T. SAULS, J. W. SCHROEDER, S. D. BROWN, G. LE TREUT, F. SI, D. LI, J. D. WANG, AND S. JUN, *Mother machine image analysis with MM3*, bioRxiv, (2019), p. 810036.
- [80] X. SHEN, S. DIAMOND, Y. GU, AND S. BOYD, *cvxgrp/dccp*, <https://github.com/cvxgrp/dccp>.
- [81] X. SHEN, S. DIAMOND, Y. GU, AND S. BOYD, *Disciplined convex-concave programming*, in 2016 IEEE 55th Conference on Decision and Control (CDC), 2016, pp. 1009–1014.
- [82] U. SIVARAJAH, M. M. KAMAL, Z. IRANI, AND V. WEERAKKODY, *Critical analysis of Big Data challenges and analytical methods*, Journal of Business Research, 70 (2017), pp. 263–286.
- [83] S. W. SLOAN, *A fast algorithm for constructing Delaunay triangulations in the plane*, Advances in Engineering Software, 9 (1987), pp. 34–55.
- [84] I. SMAL, W. NIESSEN, AND E. MEIJERING, *Bayesian tracking for fluorescence microscopic imaging*, in 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2006, pp. 550–553.
- [85] A. SMITH, J. METZ, AND P. S., *MMHelper: An automated framework for the analysis of microscopy images acquired with the mother machine*, Scientific Reports, 9 (2019), p. 10123.
- [86] C. STRINGER, T. WANG, M. MICHAELIS, AND M. PACHITARIU, *Cellpose: a generalist algorithm for cellular segmentation*, Nature Methods, 18 (2021), pp. 100–106.
- [87] S. STYLIANIDOU, C. BRENNAN, S. B. NISSEN, N. J. KUWADA, AND P. A. WIGGINS, *SuperSegger: Robust image segmentation, analysis and lineage tracking of bacterial cells*, Molecular Microbiology, 102 (2016), pp. 690–700.
- [88] H. SU, Z. YIN, S. HUH, AND T. KANADE, *Cell segmentation in phase contrast microscopy images via semi-supervised classification over optics-related features*, Medical Image Analysis, 17 (2013), pp. 746–765.

- [89] V. ULMAN, M. MAŠKA, K. E. G. MAGNUSSON, O. RONNEBERGER, C. HAUBOLD, N. HARDER, P. MATULA, P. MATULA, D. SVOBODA, M. RADOJEVIC, ET AL., *An objective comparison of cell-tracking algorithms*, Nature Methods, 14 (2017), pp. 1141–1152.
- [90] H. VANAM AND J. RETNA RAJ R, *Analysis of twitter data through big data based sentiment analysis approaches*, Materials Today: Proceedings, (2021). (in press).
- [91] T. VICAR, J. BALVAN, J. JAROS, F. JUG, R. KOLAR, M. MASARIK, AND J. GUMULEC, *Cell segmentation methods for label-free contrast microscopy: review and comprehensive comparison*, BMC Bioinformatics, 20 (2019), pp. 1–25.
- [92] Q. WANG, J. NIEMI, C.-M. TAN, L. YOU, AND M. WEST, *Image segmentation and dynamic lineage analysis in single-cell fluorescence microscopy*, Cytometry Part A: The Journal of the International Society for Advancement of Cytometry, 77 (2010), pp. 101–110.
- [93] X. WANG, W. HE, D. METAXAS, R. MATHEW, AND E. WHITE, *Cell segmentation and tracking using texture-adaptive snakes*, in IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2007, pp. 101–104.
- [94] J. WINKLE, O. A. IGOSHIN, M. R. BENNETT, K. JOSIC, AND W. OTT, *Modeling mechanical interactions in growing populations of rod-shaped bacteria*, Physical Biology, 14 (2017), p. 055001.
- [95] J. WULFF AND M. J. BLACK, *Efficient sparse-to-dense optical flow estimation using a learned basis and layers*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 120–130.
- [96] F. YANG, M. A. MACKEY, F. IANZINI, G. GALLARDO, AND M. SONKA, *Cell segmentation, tracking, and mitosis detection using temporal context*, in International Conference on Medical Image Computing and Computer-Assisted Intervention, 2005, pp. 302–309.
- [97] A. YILMAZ, O. JAVED, AND M. SHAH, *Object tracking: A survey*, ACM Computing Surveys (CSUR), 38 (2006), pp. 13–es.
- [98] Z. ZHOU, F. WANG, W. XI, H. CHEN, P. GAO, AND C. HE, *Joint multi-frame detection and segmentation for multi-cell tracking*, in Proc. International Conference on Image and Graphics, no. LNCS 11902, 2019, pp. 435–446.