2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Biological Cell Tracking via Multi-Agent Identification and Filtering

A. Tramaloni[1], A. Testa[1], S. Avnet[2], N. Baldini[2,3] and G. Notarstefano[1]

Abstract— **Understanding cellular dynamics is a fundamental topic in different biomedical applications. Nowadays, optical microscopy is one of the most used techniques to visualize cell movements. In this paper, we consider a novel cell-tracking algorithm to track multiple cells in optical microscopy videos. The proposed methodology combines two steps. First, we model cell movements and their neighboring interactions according to tailored nonlinear multi-agent systems. Then, we identify model parameters from real cellular trajectories and predict cell movements across different frames of a video. In particular, we use an Extended Kalman Filter that exploits the distributed nature of cell dynamics. Numerical experiments on videos from the Cell Tracking Challenge dataset are performed to validate the proposed method and performance metrics are shown.**

I. Introduction

Biological cell tracking within microscopy videos is a fundamental task with broad applications across diverse domains, from medical research to drug development. Among different techniques, optical microscopy is widely used to retrieve cellular images across different time instants [1]. To understand cell behaviors with their interactions, a preliminary task is to automatically extract cell trajectories over time. In this paper, we develop a multi-cell dynamical model which will be the core of a tailored cell-tracking algorithm for microscopy videos from in-vitro cultures.

We divide the literature review into two parts. First, we discuss works addressing multi-cell tracking algorithms. Then, we review system-theory approaches to cell modeling and control. Multi-Object Tracking (MOT) is a well-known problem in the literature, see, e.g., the survey [2] and reference therein. Indeed, MOT arises in several application fields, e.g., automotive and robotics. MOT algorithms have been also applied to cell videos. As an example, the Cell Tracking Challenge is a well-known benchmark for MOT algorithms on different cell settings [3]. As for MOT algorithms specifically tailored for cell tracking, the work in [4] combines convolutional neural networks (CNN) with recurrent neural networks for segmentation and tracking in microscopy images. Authors in [5] instead combine deep learning for cell segmentation and the so-called Viterbi algorithm for tracking. The work in [6] proposes an algorithm to segment and track clustered cells in time-lapse fluorescent microscopy. The study [7] proposes a multi-feature fusion re-tracking method using a "Faster R-CNN" for detection. In [8] instead, authors use a graph neural network approach to model the entire time sequence of cells as a direct graph. The work in [9] proposes a single CNN for simultaneous cell segmentation and tracking to predict cell embeddings. A different approach is presented in [10] in which cell tracking is performed based on global spatiotemporal data association. In [11], authors perform cell segmentation and tracking by extending to the so-called coupled-active-surfaces algorithm. In [12] authors propose a variational joint local-global optical flow technique. However, these works do not exploit dynamic modeling features of the cell behaviors. In our approach, we combine classical filtering methodologies with tailored multi-agent models to take advantage of the dynamic interactions among the cells in the tracking. As for theoretical approaches to cell and biological modeling, several works often leverage concepts from system identification [13] and control theory [14]. For instance, in the work in [15], tumor cell behaviors are represented as distributions via ordinary, partial or stochastic differential equations. However, these works do not focus on the dynamics of the single cell. The work in [16] proposes control schemes for gene regulation in cellular populations. The study in [17] considers the control of a cell population endowed with a bistable toggle switch. Authors in [18] propose a control system for cell-population control. In these works however the focus is on control schemes, while in our work we are interested in identification and tracking. A multi-scale modeling approach for cell mechanics is in [19]. However, here authors do not perform system identification. Differently from the above works, our paper proposes an identification and tracking scheme, exploiting real cell data, based on nonlinear multi-agent models that can capture generic cell behaviors and can be adapted to different cell types.

The contributions of this paper are as follows. We propose a parametric multi-agent model to predict biological cell dynamical behaviors. Leveraging modeling tools from multi-agent systems and robotics, we capture cell-to-cell interactions, e.g., cohesion, alignment, or repulsion, via suitable vector fields involving states of neighboring cells. Moreover, we include terms regarding the reciprocal sensing capabilities of the cells, like the perception range of other cells. In the biology area, these behaviors are representative of chemotaxis effects [20]. The considered model is not only able to capture the neighboring interactions among cells but also provides a framework to learn the behavior of a cell population. Indeed, we propose an ad-hoc system identifi-

This work was supported in part by the Italian Ministry of Foreign Affairs and International Cooperation, grant number BR22GR01.

[1] Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy. {andrea.tramaloni, a.testa, giuseppe.notarstefano}@unibo.it.

[2] Department of Biomedical and Neuromotor Sciences, University of Bologna, Bologna, Italy. {sofia.avnet3, nicola.baldini5}@unibo.it.

[3] Biomedical Science, Technologies, and Nanobiotechnology Lab, IRCCS Istituto Ortopedico Rizzoli, Bologna, Italy. nicola.baldini@ior.it.

cation procedure to fit the parametric model on real cell trajectories. Leveraging this modeling method, we develop a novel cell-tracking algorithm based on a tailored Extended Kalman Filter (EKF) involving a multi-agent system. Central to our approach is the recognition of cells as interacting agents, rather than independent entities. Consequently, we redefine the state representation within the EKF framework to reflect the collective state of the entire cell population. This allows for more precise tracking of cellular ensembles, accounting for cell-to-cell interactions. Also, this approach applies to different in-vitro cell experiments.

The paper unfolds as follows. Section II introduces the considered cell-tracking problem and the elements that characterize multi-object tracking algorithms. Section III details the multi-agent model design and the parameter identification strategy. In Section IV, we present our novel cell-tracking algorithm. Section V provides details on the dataset used in our study and numerical experiments to show the effectiveness of the proposed algorithm through standard tracking metrics.

II. Problem Formulation

In this work, we consider a cell-tracking problem in a biological scenario in which the goal is to track multiple cells in a 2D video sequence. Throughout the paper, we consider video sequences in which the number of cells N is constant. Although simplifying, this assumption allows us to model the cell population as a multi-agent system in which the focus is on modeling cell-to-cell interactions. As we detail in the next sections, these classes of models are the key components of the proposed cell-tracking strategy. The remainder of this section introduces the fundamental concepts behind cell-tracking algorithms.

The common elements that characterize cell-tracking algorithms, except for some additional components, typically include i) object detection, ii) motion prediction steps, and iii) data association. A schematic of this procedure is depicted in Figure 1.

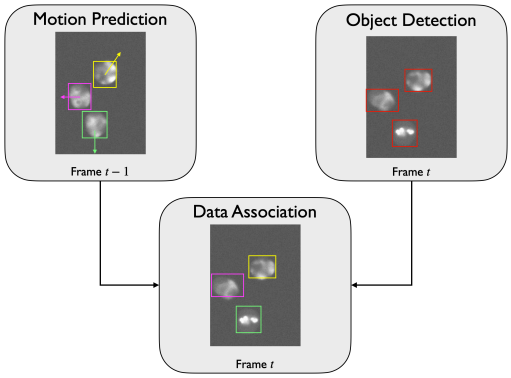

Fig. 1. Scheme of a generic multi-cell tracking algorithm

Object detection is the process of identifying and localizing cells within an image or video frame. State-of-the-art object detection algorithms typically leverage suitably trained neural networks for this task [21].

Motion prediction involves forecasting the future positions of tracked cells based on their past trajectories and motion patterns. This helps in maintaining the continuity of cell tracks and compensating for any gaps or occlusions in the detection process. Typical approaches deal with Kalman filters, particle filters, or deep learning-based predictors.

Data association is the task of linking detected cells across consecutive frames to maintain their identities over time. This involves matching detections from different frames and associating them with the same cell track. One common approach for data association is to use motion models and similarity metrics to estimate the likelihood of association between detections.

III. Cell Model Design and Identification

In this section, we introduce a multi-agent model to describe cell dynamics. The latter will be then employed in the proposed cell-tracking algorithm. The design of our model draws inspiration from a well-established line of research on bio-inspired multi-agent systems [20]. In particular, we propose a distributed model motivated by swarm-like approaches as in [22], [23]. In the following, we present a dynamical model non-linear in state and linear in learning parameters. The first property is adopted to capture complex state dynamics. The linearity in the parameters, instead, ensures a closed-form solution in the fitting problem.

A. Multi-Agent Parametric Model

The model proposed in our study is a multi-agent system with linear dependence in the parameters. In this model, the generic agent i represents a cell. Let $\mathbf{p}_i \in \mathbb{R}^2$ and $\mathbf{v}_i \in \mathbb{R}^2$ denote its position and velocity, respectively. Each agent is modeled according to discrete-time dynamics in the form

$$\begin{aligned} \mathbf{p}_i^{t+1} &= \mathbf{p}_i^t + \mathbf{v}_i^t \delta \\ \mathbf{v}_i^{t+1} &= \mathbf{v}_i^t + f_i\left(\begin{bmatrix}\mathbf{p}_i^t \\ \mathbf{v}_i^t\end{bmatrix}, \left\{\begin{bmatrix}\mathbf{p}_j^t \\ \mathbf{v}_j^t\end{bmatrix}\right\}_{j\in\mathcal{N}_i^t}, \boldsymbol{\theta}\right)\delta, \end{aligned} \tag{1}$$

where δ is a discretization step. The set $\mathcal{N}_i^t = \left\{j \neq i : \left\|\mathbf{p}_i^t - \mathbf{p}_j^t\right\| \leq R\right\}$ contains the neighbors of cell i. Here, R is a sensing radius that defines the region of space in which other cells are perceived. The vector $\boldsymbol{\theta} = [c_1, c_2, c_3]^\top$ contains constant parameters to be learned. We provide more details on their meaning in the remainder of this section. The function f_i in (1) instead is

$$\begin{aligned} f_i\left(\begin{bmatrix}\mathbf{p}_i^t \\ \mathbf{v}_i^t\end{bmatrix}, \left\{\begin{bmatrix}\mathbf{p}_j^t \\ \mathbf{v}_j^t\end{bmatrix}\right\}_{j\in\mathcal{N}_i^t}, \boldsymbol{\theta}\right) &= c_1\left(\frac{1}{|\mathcal{N}_i^t|}\sum_{j\in\mathcal{N}_i^t}\mathbf{p}_j^t - \mathbf{p}_i^t\right) + \\ + c_2\left(\frac{1}{|\mathcal{N}_i^t|}\sum_{j\in\mathcal{N}_i^t}\mathbf{v}_j^t - \mathbf{v}_i^t\right) &+ c_3\sum_{j\in\mathcal{N}_i^t}\frac{\mathbf{p}_i^t - \mathbf{p}_j^t}{\left\|\mathbf{p}_i^t - \mathbf{p}_j^t\right\|^2}, \end{aligned} \tag{2}$$

where the three constants c_1, c_2 and c_3 represent the strengths of cohesive, alignment, and separation forces, respectively. Considering R fixed (not learnable) we end up with a multi-agent dynamical model linear in $\boldsymbol{\theta}$.

B. Learning of Parameters

In this section will be described the procedure regarding the identification of parameters on the model presented in Sections III-A. The fitting problem admits a closed-form solution $\boldsymbol{\theta}^\star$. The purpose here is to find a solution that best fits a training set of real cell trajectories from in-vitro cultures. An example of trajectories used for training is in Figure 2.

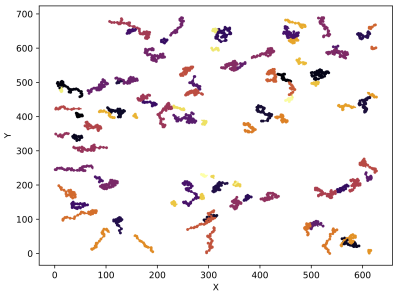

Fig. 2. Cell trajectories for identification extracted from a dataset video

Based on that, suppose having a dataset containing cell population trajectory samples. More precisely, we are supposed to know $\mathbf{p}_i^t \ \forall i, t$ and to compute $\mathbf{v}_i^t$ with some differentiation method. Taking into consideration the dynamics in equation (1) and imposing sensing radius R to be fixed (not learnable), we end up having a system with a linear dependence in $\boldsymbol{\theta}$. Therefore, by properly redefining the system we can construct and solve a typical linear least squares (LLS) problem that admits a closed-form solution. The solution to our LLS problem is

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \ \|\mathbf{Y} - \mathbf{D}\boldsymbol{\theta}\|^2 \tag{3}$$

in which

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 \\ \vdots \\ \mathbf{D}_N \end{bmatrix} \in \mathbb{R}^{(2TN)\times p}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix} \in \mathbb{R}^{2TN}$$

represent the design and observation matrices, respectively. Then, for all the cells $i \in \{1, \dots, N\}$, we define their observations as

$$\mathbf{y}_i^t = \frac{\mathbf{p}_i^{t+1} - \mathbf{p}_i^t}{\delta} - \mathbf{v}_i^{t-1},$$
$$\mathbf{Y}_i = \begin{bmatrix} \mathbf{y}_i^0 \\ \vdots \\ \mathbf{y}_i^{T-1} \end{bmatrix} \in \mathbb{R}^{2T}.$$

Instead, the matrices $\mathbf{D}_i$ are in the form

$$\mathbf{D}_i = \begin{bmatrix} d_1^0 & d_2^0 & d_3^0 \\ \vdots & \vdots & \vdots \\ d_1^{T-1} & d_2^{T-1} & d_3^{T-1} \end{bmatrix} \delta,$$

with entries

$$d_1^t := \frac{1}{|\mathcal{N}_i^t|} \sum_{j \in \mathcal{N}_i^t} \mathbf{p}_j^t - \mathbf{p}_i^t \qquad d_2^t := \frac{1}{|\mathcal{N}_i^t|} \sum_{j \in \mathcal{N}_i^t} \mathbf{v}_j^t - \mathbf{v}_i^t$$
$$d_3^t := \sum_{j \in \mathcal{N}_i^t} \frac{\mathbf{p}_i^t - \mathbf{p}_j^t}{\left\|\mathbf{p}_i^t - \mathbf{p}_j^t\right\|^2}.$$

Thus, the closed-form solution of (3) that best fits a single sample (video) of trajectories is

$$\boldsymbol{\theta}^\star = \left(\mathbf{D}^\top \mathbf{D}\right)^{-1} \mathbf{D}^\top \mathbf{Y}. \tag{4}$$

For the multi-sample (video) case, the LLS matrices can be expanded with additional data.

IV. Multi-Cell Tracking Algorithm

In this section, we detail our solution to the cell-tracking problem outlined in Section II. Our methodology combines a deep learning-based detection algorithm with a tailored Extended Kalman Filter based on interaction dynamics designed for the motion of biological cells.

The main steps of the algorithm are summarized in Algorithm 1. In particular, it starts by initializing the state guess with detections at the first frame and setting up the EKF matrices. Then, throughout each frame, the algorithm iterates through three main steps. First, it detects cells using a deep neural network. Second, it predicts the next state via the EKF prediction step personalized with interaction dynamics. Third, it associates detections to predictions and updates the state estimate through the EKF update step.

Algorithm 1 Cell Tracking with Interaction Dynamics

Initialization
get initial detections from Faster-RCNN: $\mathbf{Z}^0$,
initialize EKF: $\hat{\mathbf{X}}^{0|0} = \mathbf{Z}^0$, $\mathbf{P}^{0|0}$, $\mathbf{Q}$, $\mathbf{R}$.
for each frame t **do**
Objects Detection:
get detections from Faster-RCNN: $\mathbf{Z}^t$
EKF Prediction Step:

$$\hat{\mathbf{X}}^{t|t-1} = \hat{\mathbf{X}}^{t-1|t-1} + f\left(\hat{\mathbf{X}}^{t-1|t-1}, \boldsymbol{\theta}^\star\right)\delta \tag{5}$$
$$\mathbf{F}_{t-1} = \frac{\partial f}{\partial X}\Big|_{\hat{\mathbf{X}}^{t-1|t-1}} \tag{6}$$
$$\mathbf{P}^{t|t-1} = \mathbf{F}_{t-1}\mathbf{P}^{t-1|t-1}\mathbf{F}_{t-1}^T + \mathbf{Q} \tag{7}$$

Associate Detections to Predictions:
compute $\mathbf{S}^t$ and then $\mathbf{A}^t$
reorder detections IDs in $\mathbf{Z}^t$ based on $\mathbf{A}^t$
EKF Update Step:

$$\mathbf{K}^t = \mathbf{P}^{t|t-1}\mathbf{H}^\top(\mathbf{H}\mathbf{P}^{t|t-1}\mathbf{H}^\top + \mathbf{R})^{-1} \tag{8}$$
$$\hat{\mathbf{X}}^{t|t} = \hat{\mathbf{X}}^{t|t-1} + \mathbf{K}^t(\mathbf{Z}^t - \mathbf{H}\hat{\mathbf{X}}^{t|t-1}) \tag{9}$$
$$\mathbf{P}^{t|t} = (\mathbf{I} - \mathbf{K}^t\mathbf{H})\mathbf{P}^{t|t-1} \tag{10}$$

end for

We now detail all the steps in the algorithm.

A. Object Detection

The object detection phase serves as the initial step in our algorithm. In particular, its objective is to identify and localize cells within consecutive frames of the input video. For this step, we employ Faster R-CNN [24], one of the state-of-the-art deep learning-based detection algorithms. The latter returns a list of rectangular bounding boxes containing cells in a certain frame. Note that the entire process of object detection is performed independently for each frame in the input video. Therefore, the cell observations are specific to individual frames, meaning that Faster R-CNN output measurements possess random identity labels. Furthermore, we want to highlight that all subsequent quantities and computations within our algorithm are directly connected to the pixel coordinates of detected cells in the image. This ensures that the tracking and estimation processes maintain consistency with the spatial information provided by the object detection step. Formally, the object detection step returns, at each frame, a list of vectors in the form $\mathbf{z}_i = \left[\mathbf{p}_i^\top, s_i, r_i\right]^\top$, where $\mathbf{p}_i$ represents the position of the cell in the frame, while $s, r \in \mathbb{R}$ are the areas and the aspect ratio of each bounding box. Associated with each bounding box, the detection step also returns a set of labels ℓ_i associated with the i^{th} cell.

B. Motion Prediction

At each time t, the object detection phase yields a collection of measurement vectors $\mathbf{z}_i^t \in \mathbb{R}^m$ (cf. Section IV-A). Based on that, we can now detail the motion prediction step. Let $\mathbf{x}_i^t \in \mathbb{R}^n$ denote the real, unknown state of the i^{th} cell at time t. Formally

$$\begin{aligned} \mathbf{x}_i &= \left[\mathbf{p}_i^\top, \mathbf{v}_i^\top, s_i, r_i, v_{s,i}\right]^\top \\ \mathbf{z}_i &= \left[\mathbf{p}_i^\top, s_i, r_i\right]^\top \end{aligned} \tag{11}$$

where $v_{s,i}$ is the area rate of change of the bounding box. Recalling that $s, r \in \mathbb{R}$ and $\mathbf{p}_i, \mathbf{v}_i \in \mathbb{R}^2$, we have $n = 7$ and $m = 4$. Let also introduce $\hat{\mathbf{x}}_i^{t|t-1} \in \mathbb{R}^n$ as the predicted state of i^{th} cell at time t, obtained after a prediction step in which the measurement $\mathbf{z}^{t-1}$ is known. Instead, $\hat{\mathbf{x}}_i^{t|t} \in \mathbb{R}^n$ represents the updated estimate through the measurement $\mathbf{z}_i^t$ obtained after an estimation update step. To retrieve the aforementioned predictions, our tracking methodology adopts an Extended Kalman Filter [25]. Using an EKF has the benefit of incorporating nonlinear dynamics in the model, and it is thus suited for capturing the complex interactions among cells. We now detail its implementation.

Our tracker operates, unlike conventional trackers that handle each object independently, on a unified state representation for the entire population of cells, but includes the distributed (sparse) structure of the dynamics. Indeed, the state function of each cell i depends only on the states of neighboring cells. We define the new "swarm" state $\mathbf{X}^t$ and its relative measurement $\mathbf{Z}^t$ as

$$\begin{aligned} \mathbf{X}^t &= [{\mathbf{x}_1^t}^\top, \dots, {\mathbf{x}_N^t}^\top]^\top \in \mathbb{R}^{Nn}, \\ \mathbf{Z}^t &= [{\mathbf{z}_1^t}^\top, \dots, {\mathbf{z}_N^t}^\top]^\top \in \mathbb{R}^{Nm}. \end{aligned}$$

Similarly, we denote by $\hat{\mathbf{X}}^{t|t-1}$ and $\hat{\mathbf{X}}^{t|t}$ the stacks of predicted and updated estimates states, respectively.

According to (11), the measurement model is a standard linear algebraic relation of the type

$$\mathbf{z}_i^t = \mathbf{H}_i \mathbf{x}_i^t + \mathbf{w}_z, \qquad i = 1, \dots, N \tag{12}$$

in which $\mathbf{H}_i \in \mathbb{R}^{m \times n}$ represents the observation matrix (supposed constant and equal for all the measurements) and $\mathbf{w}_z \in \mathbb{R}^m$ the Gaussian measurement noise with zero mean and covariance $\mathbf{R}$. Compactly $\mathbf{Z}^t = \mathbf{H}\mathbf{X}^t + \mathbf{W}_z$, where $\mathbf{H} \in \mathbb{R}^{Nm \times n}$ and $\mathbf{W}_z \in \mathbb{R}^{Nm}$ are the stacks of observation matrices and measurement noise vectors, respectively.

The nonlinear multi-agent dynamics used to predict the i^{th} cell state are the ones introduced in Section III together with the cell shape information, i.e.

$$\begin{aligned} \hat{\mathbf{p}}_i^{t|t-1} &= \hat{\mathbf{p}}_i^{t-1|t-1} + \hat{\mathbf{v}}_i^{t-1|t-1}\delta + \mathbf{w}_p \\ \hat{\mathbf{v}}_i^{t|t-1} &= \hat{\mathbf{v}}_i^{t-1|t-1} + f_i\left(\bar{\mathbf{x}}_i, \{\bar{\mathbf{x}}_j\}_{j \in \mathcal{N}_i^t}, \boldsymbol{\theta}\right)\delta + \mathbf{w}_v \\ \hat{s}_i^{t|t-1} &= \hat{s}_i^{t-1|t-1} + \hat{v}_{s,i}^{t-1|t-1}\delta + w_s \\ \hat{r}_i^{t|t-1} &= \hat{r}_i^{t-1|t-1} + w_r \\ \hat{v}_{s,i}^{t|t-1} &= \hat{v}_{s,i}^{t-1|t-1} + w_{vs}. \end{aligned} \tag{13}$$

Here, we explicit the dynamics of each entry of $\hat{\mathbf{x}}_i^{t|t-1}$. We also define

$$\bar{\mathbf{x}}_i := \begin{bmatrix} \hat{\mathbf{p}}_i^{t-1|t-1} \\ \hat{\mathbf{v}}_i^{t-1|t-1} \end{bmatrix},$$

in which we omitted the time dependence for the sake of conciseness. The noise $\mathbf{w} := [\mathbf{w}_p^\top, \mathbf{w}_v^\top, w_s, w_r, w_{vs}]^\top \in \mathbb{R}^n$ is a Gaussian random variable with zero mean and covariance $\mathbf{Q}$. The second equation in (13) is implemented by applying the multi-agent parametric model discussed in Section III. More compactly, the dynamics in (13) can be rewritten as in (5), where $f : \mathbb{R}^{nN} \mapsto \mathbb{R}^{nN}$ suitably stacks the dynamics in (13), and $\mathbf{W}_x \in \mathbb{R}^{nN}$ stacks the noise vector $\mathbf{w}$ for all the cells. The step in (7) is the update of the so-called estimation error covariance matrix $\mathbf{P} \in \mathbb{R}^{nN \times nN}$. The equations in (8)–(10) instead perform the update steps of the EKF (cf. [25]). Here, $\mathbf{K} \in \mathbb{R}^{nN \times mN}$ is the Kalman gain matrix.

C. Data Association

The data association step bridges the gap between detected objects across consecutive frames, preserving their identities over time. Indeed, given two consecutive time frames $t-1$ and t, the i^{th} cell may be detected and associated to different labels ℓ_i^{t-1} and ℓ_i^t. At the generic time t, the EKF predictions in (5) provide the estimate of a novel bounding box $\hat{B}_i$ for the i^{th} cell. This predicted box is centered in $\hat{\mathbf{p}}_i^{t|t-1}$ and has area and aspect ratio $\hat{s}_i^{t|t-1}$ and $\hat{r}_i^{t|t-1}$, respectively. Similarly, at time t, let B_j denote the j^{th} bounding box found by the detection step. Based on that, let us introduce the similarity matrix $\mathbf{S}_t \in \mathbb{R}^{N \times N}$ at image frame t, where the single entry

$$\left[\mathbf{S}^t\right]_{ij} = \phi\left(\hat{B}_i, B_j\right) \tag{14}$$

represents a generic similarity function ϕ computed between the predicted bounding box of i^{th} cell and the detected one

of j^{th} cell at time t. In this work, we use the so-called Intersection Over Unit (IOU) metric [26], so that

$$\left[\mathbf{S}^t\right]_{ij} = \frac{|\hat{B}_i \cap B_j|}{|\hat{B}_i \cup B_j|}. \tag{15}$$

Given $\mathbf{S}^t$, the data association problem regards finding an optimal assignment matrix $\mathbf{A}^t$ at each frame of the video that maximizes the total similarity across all associations. Each entry in $\mathbf{A}^t$ indicates whether the i^{th} estimate is associated with the j^{th} detection. The matrix $\mathbf{A}^t$ is constructed by solving a linear-assignment problem. We recall that the linear-assignment problem can be solved in polynomial time leveraging, e.g., the Hungarian algorithm [27]. To reject associations with a poor similarity index, it is also possible to enforce a threshold in (14).

V. Numerical Experiments

In this section, we first introduce the dataset employed in our experiments. Then, we present the results obtained by applying our proposed cell-tracking algorithm.

A. Dataset

The dataset considered in this study originates from Cell Tracking Challenge, a publicly available repository housing annotated videos of moving cells. These videos feature different mono-culture scenarios, capturing the dynamics of specific cell types. The recordings encompass 2D time-lapse sequences capturing cell movement on or within a substrate, employing various microscopy techniques such as bright field, phase contrast, and differential interference contrast.

For each mono-culture scenario, the dataset typically consists of two videos supplemented with reference annotations about cell tracks and segmentations (serving as gold standards for evaluation). These videos provide the necessary reference annotations that are essential for assessing the performance of tracking algorithms. Indeed, given the segmentations, it is possible to obtain the ground-truth bounding boxes. Within the dataset, other videos are also present, but they are given as raw image sequences without accompanying annotations. Thus, we preferred not to use them. For our experiments, we selected sequences depicting the movement of HL60 tumor cell nuclei (stained with Hoechst dye). HL60 is a commonly used cell line in biomedical research, particularly in studies related to leukemia. A sample frame is shown in Figure 3.

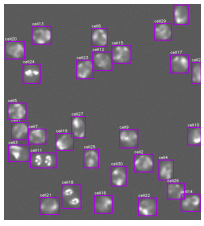

Fig. 3. Example of frame with with HL60 cells. Purple rectangles represent the bounding boxes computed leveraging ground truth segmentations

B. Experimental Results

In order to run experiments on the dataset described in Section V-A, we implemented the proposed methodology leveraging the Python3 programming language. In particular, we picked one of the two videos of the HL60 dataset to train the model in Section III. To find the optimal parameter vector as in (3) we made use of the standard linear algebra tools from NumPy library. As for the implementation of Faster R-CNN, we leveraged the TensorFlow toolbox. As for the initialization of Algorithm 1, we proceeded as follows. We set $\mathbf{P}^{0|0} = \mathbf{Q} = 100 \cdot I_{nN}$ and $\mathbf{R} = I_{mN}$, where I_q denotes the identity matrix of size q. In our paper, we are interested in investigating the performance of our tracker, rather than our detector. Thus, we ran experiments using the bounding-box annotations from the Cell Tracking Challenge dataset, in which each box is represented by its two extreme corners. The conversion with the measurement in (11) is univocal. To simulate errors in the measurement system, we added to each annotation a random noise with zero mean and covariance $2 \cdot I_4$. In Figure 4, we report a set of snapshots, taken at consecutive frames, from our numerical experiments on a different video of HL60 cells with 15 frames. For each frame t, we depict both the bounding boxes predicted by our algorithm and the real ones.

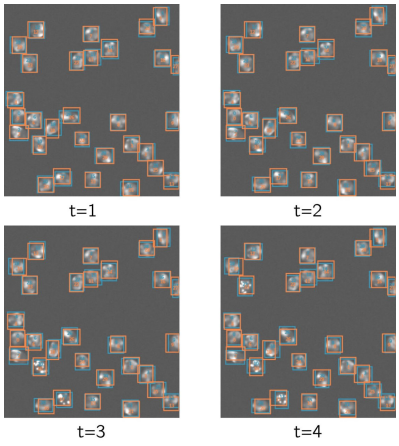

Fig. 4. Algorithm 1 simulation: predicted (orange) and ground-truth (blue) bounding boxes at consecutive time frames

We also compared our scheme against a baseline framework in multi-object tracking, namely the Simple Online Real Time Tracking (SORT) algorithm [28]. SORT employs Faster-RCNN for detection, Intersection over Union (IoU) for similarity computation, the Hungarian algorithm for data association, and a linear Kalman filter with single-integrator dynamics for motion prediction. We initialized SORT with the same matrices of Algorithm 1. The metrics that we chose to benchmark the schemes are the Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP) [29]. MOTA depends on the number of misses, false positives, and mismatches. MOTP instead compares the predicted bounding boxes against the real ones. Table I summarizes the results, showcasing the effectiveness of our

approach. Indeed, a perfect MOTA is achieved, indicating the algorithm's capability to accurately track cells over time. As indicated by the MOTP metric, the proposed tracking model is able to provide better predictions with respect to SORT.

TABLE I
PERFORMANCE COMPARISON

Algorithm	Motion prediction	MOTA	MOTP
SORT	linear KF	88.30%	89.65%
Alg. 1	EKF with model III-A	100%	92.56%

VI. CONCLUSIONS

In this paper, we developed a novel algorithm to track cells in optical microscopy videos. To this end, we modeled the cell population as a multi-agent system. Leveraging real videos from in-vitro experiments, we applied system identification techniques to learn a set of parameters weighting cell-to-cell interactions. The considered model is the building block of a cell tracking algorithm, that implements an EKF based on such a multi-agent model. To validate the design, we provided results on the Cell Tracking Challenge benchmark. Future developments include conducting additional experiments, employing co-culture systems to enhance model complexity, and making a more in-depth comparison with existing methodologies to better mimic physiological interactions, such as those between bone cancer cells and stromal cells in the tumor microenvironment.

REFERENCES

[1] E. J. Sutton, T. D. Henning, B. J. Pichler, C. Bremer, and H. E. Daldrup-Link, "Cell tracking with optical imaging," *European radiology*, vol. 18, pp. 2021–2032, 2008.

[2] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, and T.-K. Kim, "Multiple object tracking: A literature review," *Artificial intelligence*, vol. 293, p. 103448, 2021.

[3] M. Maška, V. Ulman, P. Delgado-Rodriguez, E. Gómez-de Mariscal, T. Nečasová, F. A. Guerrero Peña, T. I. Ren, E. M. Meyerowitz, T. Scherr, K. Löffler, *et al.*, "The cell tracking challenge: 10 years of objective benchmarking," *Nature Methods*, pp. 1–11, 2023.

[4] Y. Chen, Y. Song, C. Zhang, F. Zhang, L. O'Donnell, W. Chrzanowski, and W. Cai, "Celltrack r-cnn: A novel end-to-end deep neural network for cell segmentation and tracking in microscopy images," in *IEEE International Symposium on Biomedical Imaging*, pp. 779–782, 2021.

[5] D. E. Hernandez, S. W. Chen, E. E. Hunter, E. B. Steager, and V. Kumar, "Cell tracking with deep learning and the viterbi algorithm," in *IEEE International Conference on Manipulation, Automation and Robotics at Small Scales*, pp. 1–6, 2018.

[6] W. Tarnawski, V. Kurtcuoglu, P. Lorek, M. Bodych, J. Rotter, M. Muszkieta, Ł. Piwowar, D. Poulikakos, M. Majkowski, and A. Ferrari, "A robust algorithm for segmenting and tracking clustered cells in time-lapse fluorescent microscopy," *IEEE journal of biomedical and health informatics*, vol. 17, no. 4, pp. 862–869, 2013.

[7] H. Hu, L. Zhou, Q. Guan, Q. Zhou, and S. Chen, "An automatic tracking method for multiple cells based on multi-feature fusion," *IEEE Access*, vol. 6, pp. 69782–69793, 2018.

[8] T. Ben-Haim and T. R. Raviv, "Graph neural network for cell tracking in microscopy videos," in *European Conference on Computer Vision*, pp. 610–626, Springer, 2022.

[9] K. Löffler and R. Mikut, "Embedtrack—simultaneous cell segmentation and tracking through learning offsets and clustering bandwidths," *IEEE Access*, vol. 10, pp. 77147–77157, 2022.

[10] R. Bise, Z. Yin, and T. Kanade, "Reliable cell tracking by global data association," in *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pp. 1004–1010, IEEE, 2011.

[11] O. Dzyubachyk, W. A. Van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE transactions on medical imaging*, vol. 29, no. 3, pp. 852–867, 2010.

[12] F. Boukari and S. Makrogiannis, "Automated cell tracking using motion prediction-based matching and event handling," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 17, no. 3, pp. 959–971, 2018.

[13] P. Z. Marmarelis and K.-I. Naka, "Identification of multi-input biological systems," *IEEE Transactions on Biomedical Engineering*, no. 2, pp. 88–101, 1974.

[14] I. Ruolo, S. Napolitano, D. Salzano, M. di Bernardo, and D. di Bernardo, "Control engineering meets synthetic biology: Foundations and applications," *Current Opinion in Systems Biology*, vol. 28, p. 100397, 2021.

[15] H. Enderling and M. AJ Chaplain, "Mathematical modeling of tumor growth and treatment," *Current pharmaceutical design*, vol. 20, no. 30, pp. 4934–4940, 2014.

[16] V. Martinelli, D. Salzano, D. Fiore, and M. di Bernardo, "Multicellular pi control for gene regulation in microbial consortia," *IEEE Control Systems Letters*, vol. 6, pp. 3373–3378, 2022.

[17] D. Fiore, D. Salzano, E. Cristòbal-Cóppulo, J. M. Olm, and M. di Bernardo, "Multicellular feedback control of a genetic toggle-switch in microbial consortia," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 151–156, 2020.

[18] X. Ren, A.-A. Baetica, A. Swaminathan, and R. M. Murray, "Population regulation in microbial consortia using dual feedback control," in *IEEE Conference on Decision and Control*, pp. 5341–5347, 2017.

[19] T. Beyer and M. Meyer-Hermann, "Multiscale modeling of cell mechanics and tissue organization," *IEEE Engineering in Medicine and Biology Magazine*, vol. 28, no. 2, pp. 38–45, 2009.

[20] H. Oh, A. R. Shirazi, C. Sun, and Y. Jin, "Bio-inspired self-organising multi-robot pattern formation: A review," *Robotics and Autonomous Systems*, vol. 91, pp. 83–100, 2017.

[21] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey," *Neurocomputing*, vol. 381, pp. 61–88, 2020.

[22] H. Sayama, "Swarm chemistry," *Artificial life*, vol. 15, no. 1, pp. 105–114, 2009.

[23] M. Eyiyurekli, L. Bai, P. I. Lelkes, and D. E. Breen, "Chemotaxis-based sorting of self-organizing heterotypic agents," in *ACM Symposium on Applied Computing*, pp. 1315–1322, 2010.

[24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[25] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[26] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019.

[27] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[28] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *IEEE international conference on image processing*, pp. 3464–3468, 2016.

[29] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.