



Associating approximate paths and temporal sequences of noisy detections: Application to the recovery of spatio-temporal cancer cell trajectories



Matthias Dorfer^{a,1}, Tomáš Kazmar^{a,1,2}, Matěj Šmíd^{a,3}, Sanchit Sing^a, Julia Kneißl^c,
Simone Keller^c, Olivier Debeir^b, Birgit Luber^c, Julian Mattes^{a,*}

^a Knowledge-Based Vision Systems, Software Competence Center Hagenberg GmbH, Softwarepark 21, Hagenberg 4232, Austria

^b LISA-image, Université Libre de Bruxelles, Belgium

^c Institut für Allgemeine Pathologie und Pathologische Anatomie, Technische Universität München, Germany

ARTICLE INFO

Article history:

Received 3 June 2014

Revised 15 January 2015

Accepted 14 March 2015

Available online 29 April 2015

Keywords:

Tracking

Graph

Shortest path

Cancer cell

Phase-contrast microscopy

ABSTRACT

In this paper we address the problem of recovering spatio-temporal trajectories of cancer cells in phase contrast video-microscopy where the user provides the paths on which the cells are moving. The paths are purely spatial, without temporal information. To recover the temporal information associated to a given path we propose an approach based on automatic cell detection and on a graph-based shortest path search. The nodes in the graph consist of the projections of the cell detections onto the geometrical cell path. The edges relate nodes which correspond to different frames of the sequence and potentially to the same cell and trajectory. In this directed graph we search for the shortest path and use it to define a temporal parametrization of the corresponding geometrical cell path. An evaluation based on 286 paths of 7 phase contrast microscopy videos shows that our algorithm allows to recover 92% of trajectory points with respect to the associated ground truth. We compare our method with a state-of-the-art algorithm for semi-automated cell tracking in phase contrast microscopy which requires interactively placed starting points for the cells to track. The comparison shows that supporting geometrical paths in combination with our algorithm allow us to obtain more reliable cell trajectories.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Despite of being a research topic for several decades, in many application scenarios automated object tracking algorithms still suffer from frequent loss of target or fail to meet the level of precision in the resulting trajectories required by the user (Kan et al., 2011; Maška et al., 2014). In such situations user interaction to support the algorithm is necessary. Different options are available: For instance, the user detects where the trajectories provided by the algorithm are erroneous and edits them at these points interactively. Or, the user provides some prior information such as seed points allowing the algorithm to behave more accurately and robustly. Moreover, a combination of both options might be the adequate strategy.

In this paper we face a tracking problem with specific prior information: The geometrical paths (without temporal information) of the objects to be tracked are provided by the user. The paths are drawn on the screen while playing the video sequence in the background. Depending on the number of failures which are expected when using an automated approach this can be a reasonable alternative to the post-processing of the paths by manually correcting the spatio-temporal associations of the trajectories. Technically, the problem can be formulated (cf., Section 1.2) as the recovery of the spatio-temporal parametrization – answering the question: on which position of its path is a given object located at a certain point in time?

In particular, we consider moving objects which are represented for a given point in time by a single point in space. This problem has various different applications in biology (cell tracking (Sbalzarini and Koumoutsakos, 2005), fluorescent protein-tracking (Carter et al., 2005; Mallik et al., 2004), tracking of sub-cellular structures (Carter et al., 2005; Catlett and Weisman, 2000), etc.), physics (particle tracking (YH et al., 2014)), medicine (tracking characteristic points of medical objects (Rahim et al., 2011)), traffic & surveillance (tracking of pedestrians Snidaro et al. (2004), cars Robert (2010), etc.), sports, etc.

* Corresponding author. Tel.: +4372363343835; fax: +4372363343888.

E-mail address: Julian.Mattes@scch.at, Julian.Mattes@vidiatech.at (J. Mattes).

¹ M. Dorfer and T. Kazmar contributed equally to this work.

² Current address: IMP – Research Institute of Molecular Pathology, Austria.

³ Current address: Center for Machine Perception, Czech Technical University in Prague, Czech Republic.

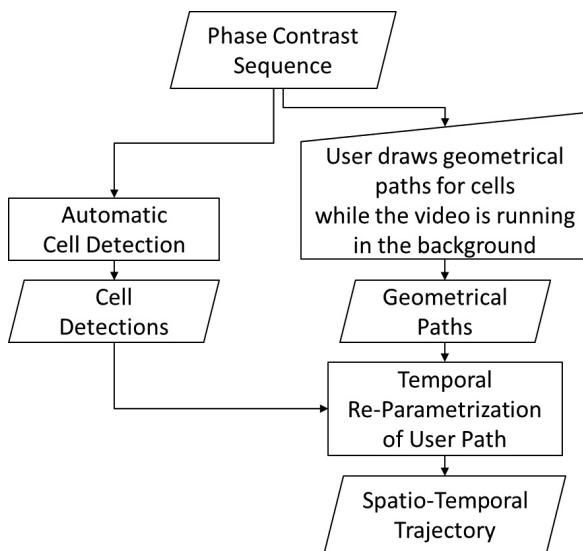


Fig. 1. General overview of proposed tracking procedure. (input/output: parallelogram, user input: trapezoid, automatic processing: rectangle).

In our application, the objects are cultivated cancer cells moving on a matrix-coated glass-bottom culture dish and filmed using phase-contrast microscopy. All biologically relevant cells on a film have to be tracked in order to describe their motion by a number of quantitative values with the purpose of investigating the effect of different drugs on the motile behavior of the cells (Decaestecker et al., 2007).

Our approach proceeds in two steps (cf., Fig. 1). First, the user annotates the geometrical paths (without temporal information) on the two-dimensional scene. Compared to manual frame-by-frame tracking this allows to play the film in the background when annotating the path and to place for each path only a small number of points into the scene. Namely, it allows that the position of a cell can still be marked when it moved already further on. Therefore, annotating paths becomes notably less time consuming than manual frame-by-frame tracking, however, the temporal information for the points on the path is lost.

The resulting paths allow us to calculate averaged temporal parameters, such as average cell speed, but parameters where the local change in the spatio-temporal behavior is relevant (e.g., maximum speed, maximum dispersion, or displacement entropies) (Costa et al., 2005; Meijering et al., 2012) cannot be determined. Knowing the temporal course of the effect of a drug treatment is necessary yet, in order to differentiate different mechanisms of molecular regulation (such as RNA transcription or protein phosphorylation). Hence, in a second step, we propose an automated procedure for recovering the fully spatio-temporally parametrized trajectories from approximate geometrical paths. We consider this step as a reasonable application of automation for gaining relevant information while saving user time.

One aim of our approach for defining such an automated procedure is to provide a generic framework based on a modular concept where modules can be exchanged in order to apply the algorithm in different application scenarios. Therefore, we apply a cell detector (Kazmar et al., 2010) as an independent module to every frame of the image sequence. Afterwards, we try to automatically associate the so obtained points of detection with the corresponding geometrical path (hence, corresponding detections, i.e., detections of the same cell in different frames are implicitly associated).

The problem of associating a geometrical path to a temporal sequence of noisy detections has also applications outside the field of biology. For instance, such an association is required as well in map-matching for car navigation, where the positions detected by

the global positioning system (GPS) have to be matched to a road map (Scott, 2004).

1.1. Related work

In this section we review related work in the field of cell tracking. Following Li et al. (2008), Maška et al. (2014) or Magnusson et al. (2014), we distinguish “tracking by detection” and “tracking by model evolution”. A combination of both approaches has been proposed as well Li et al. (2008).

Tracking by detection divides the task of cell tracking into two sub-steps (Breitenstein et al., 2009; Sbalzarini and Koumoutsakos, 2005). First, the cells are detected in the individual frames of the considered image sequence. Second, detections in successive frames are linked. For the second step—the data association task—the detected object is often reduced to a point position. This makes the cell tracking problem similar to particle tracking (Chenouard et al., 2013; Jaqaman et al., 2008) where the spatial extension of the objects is usually of minor importance. A review of both, cell tracking and tracking of sub-cellular particles, has been provided by Rohr et al. (2010).

Depending on the application the problem of linking detections can be approached by global optimization with respect to all frames of the sequence (Bise et al., 2011; Magnusson et al., 2014; Padfield et al., 2008) and/or by linking all cells or particles in the respective frames simultaneously (Al-Kofahi et al., 2006; Li et al., 2010). The former setting requires that the whole image sequence is available which is usually the case for biological applications. The latter problem formulation results in multi-target tracking and assumes that the sharing of detections is forbidden or at least penalized. Multi-target tracking has been formulated as an integer linear program which can be relaxed to a linear program (Jiang et al., 2007; Zhang et al., 2008).

Jaqaman et al. (2008) tackle detection linking by a multi-target tracking approach consisting of a two level spatially global optimization. In the first step detections in consecutive frames are related. Once these temporally local assignments are established, they are used as a basis for gap closing as well as track segment merging and splitting. This second step is addressed in a spatial and temporal global optimal manner. For the sake of complexity reduction they realize the first step (finding track segments) with a greedy approach.

For taking all frames of the image sequence into account during detection linking, Magnusson et al. (2014) are representing the detection as states in a state space diagram. Additional states are created for cells which appear later on (are “born later”) or have disappeared (are “dead”). Based on this state space diagram and on a scoring function defined on a set of tracks, they start from an empty set of tracks. Then, tracks are added sequentially by searching for the respective shortest path in the state space diagram. The shortest path can be found with linear complexity using the Viterbi algorithm thanks to the “trellis structure” of the state space diagram (Magnusson et al., 2014). The algorithm handles detection sharing by a swap operation. In order to maximize the scoring function a link in the path of one of the concerned tracks is edited and assigned to a new detection.

The authors obtained with this algorithm “outstanding performance” at ISBI 2013 Cell Tracking Challenge (Maška et al., 2014) and achieved “better performance than all other systems on all of the challenge datasets” (Magnusson et al., 2014) at ISBI 2014 Cell Tracking Challenge. The results of 2013 have been published by Maška et al. (2014), the results of the ISBI 2014 Cell Tracking Challenge are available on the ISBI homepage.⁴

However, the results which can be achieved with this algorithm depend heavily on the quality of the available detections which in

⁴ http://www.codesolorzano.com/celltrackingchallenge/Cell_Tracking_Challenge>Welcome.html.

turn depends on the difficulty of the detection task for the given type of data. In particular, for their evaluation [Magnusson et al. \(2014\)](#) did not allow to skip frames in order to deal with missing detections as in either of the datasets they are processing “there is no problem with missed detections”. The algorithm itself would allow frame skipping. For the ISBI 2014 Cell Tracking Challenge two sequences from phase contrast microscopy have been added. One sequence contains a densely distributed cell population where the individual cells have a homogeneous and rather simple appearance. The second sequence contains cells with complex and varying appearance but sufficiently distant from each other avoiding collisions or the formation of agglomerations. The most critical situation, however, for cell detectors ([Kazmar et al., 2010](#)) is the co-occurrence of both problems: densely distributed cells forming agglomerations together with a complex and varying appearance of the individual cells.

Avoiding the problem of incomplete and noisy detections together with false positive detections, tracking by model evolution propagates a model of the cell from one frame of the sequence to the next one. This allows to take the information of the previous frames into account when fitting the model into the following frame. The template model may incorporate gray level or shape information as used in parametric or geometric active contour models ([Debeir et al., 2005; Maska et al., 2013; Padfield et al., 2008; Zimmer et al., 2002](#)).

The model of [Debeir et al. \(2005\)](#) is tailored to track cells imaged with phase contrast microscopy. It uses an assembly of mean-shift kernels associated with parameters allowing a tuning to the appearance of the cells in the phase contrast images (for details see [Section 3.2](#)). A software framework implementing this model for biologists as an intended user group has been proposed recently by [Cordelières et al. \(2013\)](#). Whereas [Debeir et al. \(2005\)](#) are only interested in tracking the cell center [Padfield et al. \(2008\)](#) are interested in the contour as well and formulated a spatio-temporal segmentation problem. [Liu et al. \(2012\)](#) focused on mitosis segmentation including a temporal segmentation of different mitosis stages. They use a semi-Markov model and a hidden conditional random field classifier for the identification and further segmentation of mitosis events.

Tracking by model evolution is particularly appropriate if the changes in appearance, shape or position of the cell from one frame to the next are not too large. Difficulties may occur if two touching cells are starting to separate. An additional procedure or user interaction is required for determining the initial position of the cells, for cells entering or leaving (when tracking from the last frame to the first one) the frame during recording. If a cell is lost it is usually not recovered by the same track.

[Li et al. \(2008\)](#) combine tracking by model evolution with the concept of tracking by detection. A detector allows them to deal with determining initial cell position, with entering and leaving cells or with reappearing cells which have been lost during tracking by model evolution. In the latter case a track linker allows joining of partial tracks.

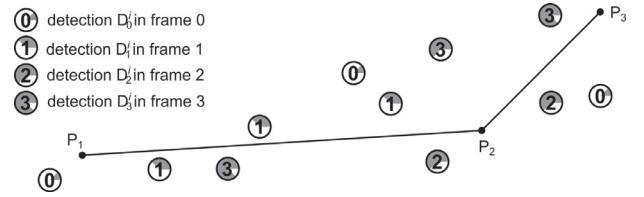
1.2. Problem formulation and definitions

Given an interval $I = [0, L]$ and a (topological) path $P = P(s)$ defined by

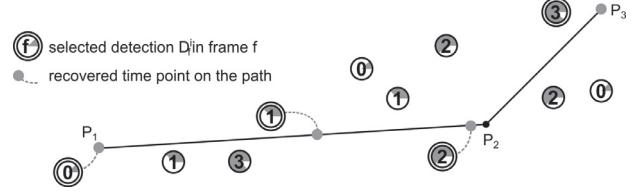
$$P : \begin{cases} [0, L] \rightarrow \mathbb{R}^2 \\ s \mapsto P(s) \end{cases}, \text{ where } P \text{ is continuous,}$$

we define a (geometrical) path P mathematically as the image of I under P : $P = P(I)$.

When we speak about the topological path P associated to the path P we consider that P is *parametrized by its arc length* which assumes the object is moving from $P(0)$ to $P(L)$ with unit speed. Hence, L is the arc length of the whole path P .



(a) Input for the proposed algorithm: A set of detections $\{D_f^i\}_i$ in different frames along with path P and its sample points.



(b) Relevant detections bring the temporal information to the path.

Fig. 2. Overview of problem statement: Data fusion of the geometric input path (no temporal information) with detections in the respective frames. The detections are taken into account to bring temporal information to the geometric path. (a) Input for the proposed algorithm: A set of detections $\{D_f^i\}_i$ in different frames along with path P and its sample points. (b) Relevant detections bring the temporal information to the path.

In contrast, we define a *(spatio-temporal) trajectory*, of an object as the topological path $T = T(t)$,

$$T : \begin{cases} [0, F] \rightarrow \mathbb{R}^2 \\ t \mapsto T(t) \end{cases}, \text{ } T \text{ continuous,}$$

where t represents a point in time which is mapped to the object's location $T(t)$ at time t .

In our application, F is the elapsed time of the video and the interval $[0, F]$ is given by discrete points in time, represented by the frame numbers $0, \dots, F$. Moreover, in our case the (geometrical) path is obtained from discrete samples $P = (P_1, P_2, \dots, P_N)$ by (linear) interpolation (we have $P_1 = P(0)$ and $P_N = P(L)$). Thereby, the number N of discrete sample points on P (set by the user) is usually much smaller than the number of frames F (cf., [Section 3.1](#)).

For an object with trajectory T which is moving on path P , this relationship between T and P is represented by the existence of a strictly increasing, surjective function ϕ with

$$\phi : \begin{cases} [0, F] \rightarrow [0, L] \\ t \mapsto \phi(t) \end{cases} \text{ such that } T(t) = P(\phi(t)).$$

Hence, $\phi(0) = 0$, $\phi(F) = L$ and $\phi(t) = s$ is the (arc) length of the already covered part of the path/trajectory at time t .

Problem statement. We denote the cell detections in frame f by a set of points $\{D_f^i\}_i = \{D_f^i \in \mathbb{R}^2 \mid i = 0, \dots, K_f\}$, where K_f is the number of detections in frame f (see [Fig. 2\(a\)](#)).

Given are a number of (geometrical) paths and a temporal sequence of noisy detections $(\{D_f^i\}_{f=0}^F)$. The respective function ϕ (and hence the trajectory T) should be recovered by associating the detections D_f^i with their respective path P under the following constraints:

- A1** Detections assigned to path P should be close to the estimated position on P .
- A2** At most one detection per frame is associated to a single path P . This leads to a *track of detections*. Consecutive detections in this track should be close.
- A3** Sharing of detections should be allowed/forbidden/penalized among trajectories (according to the application).

- A4** A path P should be passed through without change in direction (preserved temporal order of trajectory points on P) from the start of the path to its end.
A5 A trajectory should begin in P_1 and end in P_N .

In practice, we have to recover the function ϕ only in discrete points corresponding to the frames as shown in Fig. 2(b). We call (**A4**) *order constraint*. It should ensure the monotonicity of ϕ . However, we have to formulate this constraint as a soft constraint as the path is given only approximately and, especially in situations of little motion, the temporal order of detections on P can contradict the order constraint.

1.3. Different settings of the problem

The problem can be formulated as a single or multi-target tracking problem. The choice between these formulations defines how we deal with constraint (**A3**). A multi-target tracking approach (cf., Section 1.1) allows to penalize or forbid detection sharing. In contrast, if penalizing or forbidding detection sharing is not desired, the problem can be formulated as a sequence of single target tracking problems. The resulting individual problems can then be solved independently from each other.

In this paper, we allow detection sharing and therefore solve for single trajectories individually. Detection sharing is required since cells may mutually occlude each other, especially, during mitosis. In such cases only a single detection is expected although belonging to different geometrical paths.

We propose our approach in the next section, followed by an experimental evaluation in Section 3 and by a discussion and conclusion in Section 4.

2. Cell tracking based on detection-graphs and shortest path search

This section describes the process of parameterizing a path P (with $P = P(s)$) in order to turn it into a spatio-temporal trajectory $P(\phi)$. We split the process into three main steps. First, we project each detection onto path P and construct a graph based on the obtained projections (Section 2.1). Second, we identify the *track of projections* (with associated detections) that most probably belongs to the respective path (Section 2.2). Third, we recover the trajectory based on this track (Section 2.2).

Fig. 3 provides an overview of the entire process. The reminder of this section will address the single computational steps in detail.

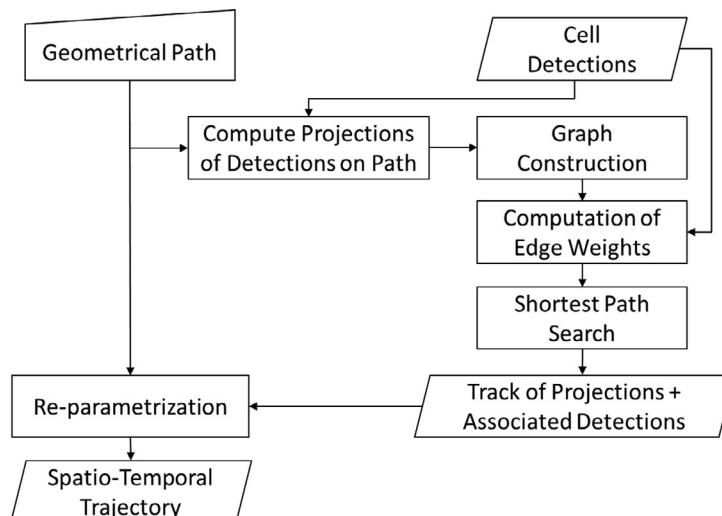


Fig. 3. Overview of spatio-temporal re-parametrization (input/output: parallelogram, user input: trapezoid, automatic processing: rectangle).

2.1. Graph construction

In order to obtain an optimal selection of projections and associated detections for a given geometrical path we construct a graph, in which projections of detections form the set of nodes and edges model possible transitions of a target object from one frame to a subsequent one. Once the graph is constructed we define weights on its edges and use it (see Section 2.2) to search for the track of projections as the shortest path from the beginning of the sequence towards its end. This formulation of data association allows choosing the globally best track of projections (and associated detections), not just the best single detections in individual frames.

2.1.1. Definition of projections and graph construction

For each detection D_f^i , we define a set of projections on path P . In the present case where P is a polyline, the projections, denoted by $V_f^{i,n}$, are the points closest to the detection D_f^i on the respective line segment n of P . This is shown in Fig. 4(a) and (c). The need of calculating multiple projections is motivated by the example provided in Fig. 4(c). We denote the set of all projections of D_f^i by \mathcal{V}_f^i , formally: $\mathcal{V}_f^i = \{V_f^{i,n} \mid n = 1, \dots, N - 1\}$. In order to unite the projections for all detections of a frame f into one set we define the set \mathcal{V}_f of all projections of f by: $\mathcal{V}_f = \bigcup_i \mathcal{V}_f^i$, where i runs through all detections D_f^i in frame f .

Based on the sequence of projections sets $(\mathcal{V}_f)_{f=0}^F$ we construct for every given path P a directed graph $G = (V, E)$ where V is the set of nodes and E the set of directed edges. In G each projection $V_f^{i,n}$ becomes a node in V (if not excluded during gating, cf., Section 2.1.3). We additionally add the two special nodes *start* and *end* in frame -1 and $F + 1$, respectively. The special nodes represent the beginning and the end of P . An edge $(V_f^{i,m}, V_g^{l,n})$ connecting two projections $V_f^{i,m}$, $V_g^{l,n}$, in frame f and g , is added to E if $1 \leq g - f \leq f_{\text{skip}}$ and if it is not omitted due to gating (cf., Section 2.1.3). This means, that we add only edges directed forward in time but allow skipping of a limited number f_{skip} of frames. The parameter f_{skip} is fixed for all frames and all paths of the image sequence. Fig. 4(b) shows an example of a projections graph as described above. Using this graph definition we can work with incomplete sets of detections, e.g. detectors with non-zero false-negative rates having missing detections.

2.1.2. Definition of edge weights

In the following we define the weight for each directed edge $(V_f^{i,m}, V_g^{l,n})$ in G . Ideally, the weights are set in a way that the shortest

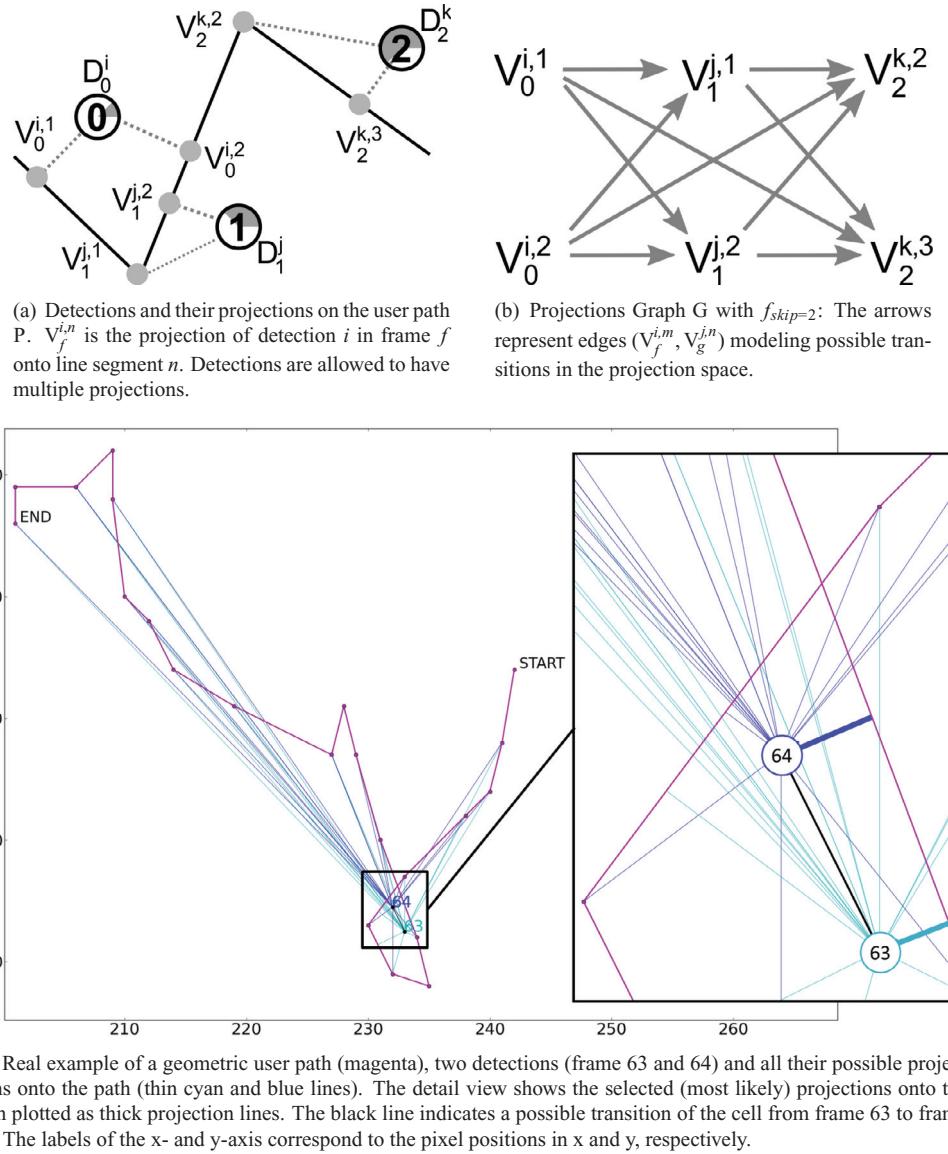


Fig. 4. Sketch of projections graph G and spatio-temporal re-parametrization. The figures show detections and their respective projections onto the geometric user path. We also show a real test example for one of the sequences introduced in Section 3.1. Sub-figure (c) illustrates that for detection “64” the projection to the line segment above on its left hand side leads to the shortest detection to path distance. However, in this case the distance on shape d_p is negative and passes through almost the whole loop present in this path. This leads to a high value of cost term C_{EP} (Eq. (3)). This contributes to the choice of the emphasized projections in the final track of projections as shown in Fig. 11(c).

path from *start* to *end* visits exactly the nodes forming the sequence of projections whose associated detections correspond to the cell belonging to user path P .

We define a signed P -distance $d_P(A, B)$ between two points A and B of P as the arc length on P from A to B . Formally, we note $A = P(s_1)$ and $B = P(s_2)$ and obtain $d_P(A, B) = \int_{s_1}^{s_2} ds = s_2 - s_1$. We have $d_P(A, B) = -d_P(B, A)$. For each edge $(V_f^{i,m}, V_g^{j,n})$ in E , we define a basic cost term C_E depending on $V_f^{i,m}$ and $V_g^{j,n}$ and on their respective detections D_f^i and D_g^j :

$$C_E = \lambda_{\text{DE}} C_{\text{DE}} + \lambda_{\text{EP}} C_{\text{EP}}, \quad (1)$$

where λ_{DE} and λ_{EP} are weights for the respective terms.

The first term penalizes projections too far from their detections and is defined by:

$$C_{\text{DE}} = \left(\|D_f^i - V_f^{i,m}\| + \|D_g^j - V_g^{j,n}\| \right) / 2 \quad (2)$$

The second term in Eq. (1) penalizes the discrepancy between the distance of the two detections compared to the P -distance of their

respective projections:

$$C_{\text{EP}} = |d_P(V_f^{i,m}, V_g^{j,n}) - \|D_f^i - D_g^j\|| \quad (3)$$

Therefore, C_{EP} implicitly penalizes breaking the order constraint (**A4**) as the P -distance becomes negative in cases of reversed order. As C_{DE} is enforcing (**A1**), C_E enables us to enforce (**A1, A4**). The final costs C assigned to the edge $(V_f^{i,m}, V_g^{j,n})$ are:

$$C = C_E + \lambda_{\text{DD}} C_{\text{DD}} / (g - f) + \lambda_{\text{skip}} C_{\text{skip}} \quad (4)$$

where $C_{\text{DD}} = \|D_f^i - D_g^j\|$ is the Euclidean distance between the detections (enforcing, together with the definition of the edges, constraint (**A2**))). $C_{\text{skip}} = g - (f + 1)$ together with its weight λ_{skip} penalizes frame skipping.

2.1.3. Gating

We apply gating (Blackman and Popoli, 1999) to V and E , which makes the graph sparse and decreases the runtime of the algorithm. In particular, for a given path we (i) ignore the nodes in the graph

associated to detections with a distance $\|D_f^i - V_f^{i,m}\| > g_{D2P}$, where g_{D2P} is a threshold. (ii) We allow only edges $(V_f^{i,m}, V_g^{j,n})$ between the nodes $V_f^{i,m}$ and $V_g^{j,n}$ not exceeding a variable distance threshold $g_{D2D}(f,g) = \min((g-f)g_{D2D}^{\min}, g_{D2D}^{\max})$, depending on two fixed values g_{D2D}^{\min} and g_{D2D}^{\max} , using the following distance terms. We keep $(V_f^{i,m}, V_g^{j,n})$ if:

- the P -distance $d_P(V_f^{i,m}, V_g^{j,n}) < g_{D2D}(f,g)$,
- the Euclidean distance $C_{DD} < g_{D2D}(f,g)$ and
- $C_{EP} < g_{D2D}(f,g)$.

2.2. Shortest path search and re-parametrization

Given the graph of projections G as described in Section 2.1 we present in this section how to obtain the desired re-parametrization of the user path P . We apply Dijkstra's algorithm (Dijkstra, 1959) in order to obtain the shortest path in the graph in form of a track of projections $V_T = (V^1, V^2, \dots, V^M)$, where M is the number of nodes in the shortest path. Thereby, we have $V^1 = P_1$ and $V^M = P_N$. Subsequently, we use the track of projections $V_T = (V^1, V^2, \dots, V^M)$ and the associated sequence of frame numbers $(0, f_2, \dots, f_{M-1}, F)$ (V^i belongs to frame f_i) to re-parameterize the geometric path P .

For each projection V^i in V_T we calculate the P -distance $d_P(0, V^i)$. We define $\phi(f_i) = d_P(0, V^i)$ for $2 \leq i \leq M-1$ and interpolate linearly in between the points in time $(0, f_2, \dots, f_{M-1}, F)$. This defines ϕ and hence concludes the conversion of a geometrical path P into a spatio-temporal trajectory $P(\phi(t))$.

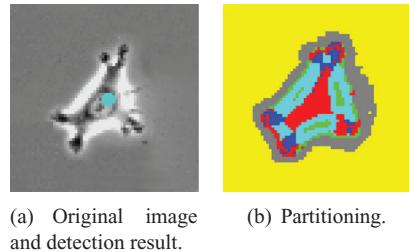
3. Experiments and discussion

We start this section by introducing our data as well as the experimental setup used to evaluate the method proposed in the previous section. We further describe an automatic cell detector which provides the detections required for re-parametrization. For a quantitative evaluation of our method we carry out two experiments on synthetic as well as on real cell data.

3.1. Data

For evaluating our graph-based approach we use seven video sequences (denoted in the following by $S_1 - S_7$) showing cells of a gastric cancer cell line. The cells were moving in a matrix coated glass-bottom culture dish and have been imaged using phase contrast microscopy with an Axiovert laser scanning microscope LSM 510 (Zeiss, Jena Germany) with lens PNF 20x/0.4 Ph2. One frame was recorded every three minutes over a period of 7 h (141 frames in total). The images have a size of 512×512 px, corresponding to $650 \times 650 \mu\text{m}^2$, and a gray scale range of 8-bit. There are approx. 40–80 cells imaged in a single frame, not all of them being tracked. The cells for tracking were selected by our biological experts to avoid taking cells into account which are undergoing cleavage or which are not present in all frames of the sequence. When selecting the cells our experts were neither aware of the proposed algorithm nor of results obtained with it or one of its parts. In total they selected 286 cells which are distributed on the sequences as follows: 32 in S_1 , 41 in S_2 , 56 in S_3 , 30 in S_4 , 49 in S_5 , 29 in S_6 and 49 in S_7 .

In addition, approximate paths in form of polyline vector overlays have been annotated by our biological experts and saved together with the respective original sequence in an LSM⁵ file. A path contains on average a set \mathcal{P} of 14 sample points in comparison to the number of 141 frames per sequence ($|\mathcal{P}| \ll F$). Finally, we manually annotated



(a) Original image and detection result.
(b) Partitioning.

Fig. 5. Result of partitioning and detection as proposed in Kazmar et al. (2010).

tracking ground truth by clicking on each of the 286 cell trajectories in each of the 141 frames of the respective sequence. As tracking (annotation) target we define the centers of the cell nuclei.

3.2. Experimental setup and technical implementation

In this section we provide details on several aspects of the technical implementation and on the parameters used in the evaluations as well as in the required automatic cell detection component. For the purpose of a comparative evaluation of the tracking results we further recall an alternative tracking method and present our evaluation measures.

3.2.1. Implementation and parameters

We implemented our graph-based algorithm proposed in Section 2 in Python using the NumPy and NetworkX libraries. For evaluation we focus in particular on our implementation of the recovery of trajectories based on the re-parametrization by projections as proposed in Section 2. We denote the re-parametrization track (and the algorithm to calculate it) in the following by G-P. In addition, we also keep the track of the associated detections denoted by T-D. This allows to investigate the accuracy of the provided user paths as discussed in Experiment 1 (cf., Section 3.3).

We tune the parameters of G-P to their best values according to prior evaluations based on synthetic data and an additional training sequence (results not shown). This leads to the parameter setting described in the following. Since, we expect missing cell detections the maximum number f_{skip} of frames allowed to skip is set to 20. f_{skip} is iteratively increased by 5 whenever no path from start to end exists in the detections graph G . In a number of cases missing detections would prevent the algorithm from finding an accurate path through the constructed graph. We are using $\lambda_{EP} = \lambda_{DE} = \lambda_{DD} = 1$ and $\lambda_{skip} = 10$. We use a value of 15 px for the gating distance g_{D2P} between a projection and its associated detection and we set $g_{D2D}^{\min} = 20$ and $g_{D2D}^{\max} = 30$.

Except for Experiment 1, where f_{skip} is set to 1, we keep this parameter settings for all evaluations presented. For the evaluation of G-P on the data described above we used it in combination with an algorithm for automatic cell detection which is described in the next section.

3.2.2. Automated cell detection

In Kazmar et al. (2010) we propose a methodology for cell detection in single phase contrast microscopy images. The method operates on two levels of abstraction. On its basis it derives a pixel wise partitioning of the input image into a predefined number of classes (can be considered as a segmentation as well). This initial step is based on a variety of texture features and supervised classification as described in the original paper in detail. For partitioning we define six distinct classes: background, membrane, nucleus, border, halo and float (cells appearing as a small white annulus). Fig. 5 shows an example partitioning for a single cell. In parallel, candidate markers are generated on the input image using a Hessian matrix based seed point detector

⁵ http://www.zeiss.com/microscopy/en_de/website/downloads/lsm-image-browser.html.

(Kazmar et al., 2010). The second level of abstraction takes the partitioning and the candidate markers produced in the first level as input. In Kazmar et al. (2010) a set of rules is defined in order to filter or join the generated markers for obtaining a single marker per cell. We replaced this rule system by a second step of supervised classification in order to learn which markers should be (1) filtered out or (2) should be joined with other markers in cases where they mark the same cell. Once this filtering and joining is trained the algorithm can be applied to arbitrary images of the same cell line recorded under similar conditions.

For the gastric cancer cells investigated in this paper, we obtain a recall and a precision of about 90%. These results are used as the detection input $\{D_f^i\}_i$ for the graph based re-parametrization proposed in Section 2.

3.2.3. Tracking with combined mean-shift processes

As a baseline for the evaluation of our approach we use a state-of-the-art method for semi-automatic cell tracking. We choose for this purpose the algorithm proposed by Debeir et al. (2005) which is especially designed for tracking in phase microscopic videos. A re-implementation of the method was recently published as an *ImageJ* based Java tracking framework by Cordelières et al. (2013). Our data includes the co-occurrence of both problems discussed for phase contrast microscopy in Section 1.1. Debeir et al. have shown that their algorithm performs well on tracking data similar to ours (Debeir et al., 2005). In this section we summarize the basic working principle of the approach. Moreover, we compare the graph-based approach proposed in Section 2 with this reference in order to analyze how supporting geometrical paths can be effectively used as an user input for cell tracking.

Briefly, the tracking algorithm is a model based approach, which fits an assembly of mean-shift kernels to the cell image. Starting from a given initial position the cell location is updated automatically for each consecutive frame. This requires an expert to place initial cell seeds on the first image of the sequence. The model itself depends on several parameters, where some of these parameters are dimensional, i.e., related to the average cell size (depending on both, cell size and acquisition resolution), and others are dimensionless (e.g., the weight ratio between the soma and the halo during the cell positions update) (Debeir et al., 2005). While the former parameters have to be adjusted with respect to the image resolution of the sequence, the latter can be considered as constant for various cell types and acquisition conditions. Therefore, the parameters adjustment has to be done only once for a specific cell type and acquisition setup. Once the setting is trained, the tracking algorithm is applied on the entire sequences. For the present application, only the dimensional parameters (i.e., the average cell halo and soma radii) required an adjustment, default values were used for other parameters. Among the tested sequences, the tracking algorithm fails for some cell trajectories, in particular when cells which are touching each other start to separate. We denote the algorithm in the following by CMS (as it is based on so-called “Combined Mean-shift Processes”).

3.2.4. Evaluation measures

In the final part of this section we introduce the methods used for comparing the tracking approaches. We implement two measures with respect to a given ground truth trajectory G_T . The focus of our evaluation is on the exact location of the cell nucleus center. This makes segmentation overlap based error measures such as introduced in the ISBI cell tracking challenge (Maška et al., 2014) impractical, although we reuse some of the concepts.

For a given image sequence and a target trajectory T we define the following measures:

- **Average error ϵ** calculating the mean of the Euclidean distances ϵ_f (in pixels) between the point $T(f)$ on the trajectory in the respective

frame f and the ground truth location $G_T(f)$ in the same frame for each frame of the sequence. Given the per frame distance $\epsilon_f(T) = \|G_T(f) - T(f)\|$, ϵ is defined as:

$$\epsilon(T) = \frac{1}{F+1} \sum_{f=0}^F \epsilon_f(T) \quad (5)$$

- **Reliability score ς_l** depending on a cut-off level l . According to Maška et al. (2014) we call the association between $T(f)$ and $G_T(f)$ a *temporal relation* and we consider it as *correctly recovered* if $\epsilon_f \leq l$. We define the *reliability score ς_l* as the ratio of the number r of correctly recovered temporal relations and the total number of frames in the sequence (see Kan et al., 2011):

$$\varsigma_l(T) = r/(F+1) \quad (6)$$

We consider a trajectory T as recovered if its reliability score $\varsigma_l(T)$ is above 95%. The reliability score ς_l reflects the situation that above a certain degree of error a trajectory is not useful for further biological evaluations and therefore should be rejected.

3.3. Experiment 1 – Precision of the paths provided by the user

When searching for the temporal re-parametrization of approximate geometrical paths in order to recover spatio-temporal trajectories the precision of the manually drawn paths is a lower bound for the precision we can achieve with our approach. In order to be able to appraise this precision in comparison with the results obtained by G-P and CMS we carried out a first experiment based on detection ground truth.

We take the ground truth positions of all trajectories available in the image sequences as synthetic detections D_f^i and consider them as the best possible result of a detector. Using this setting we compute G-P as well as T-D and compare it with CMS. CMS has apart from manually placed initial seed points for every target cell no prior information

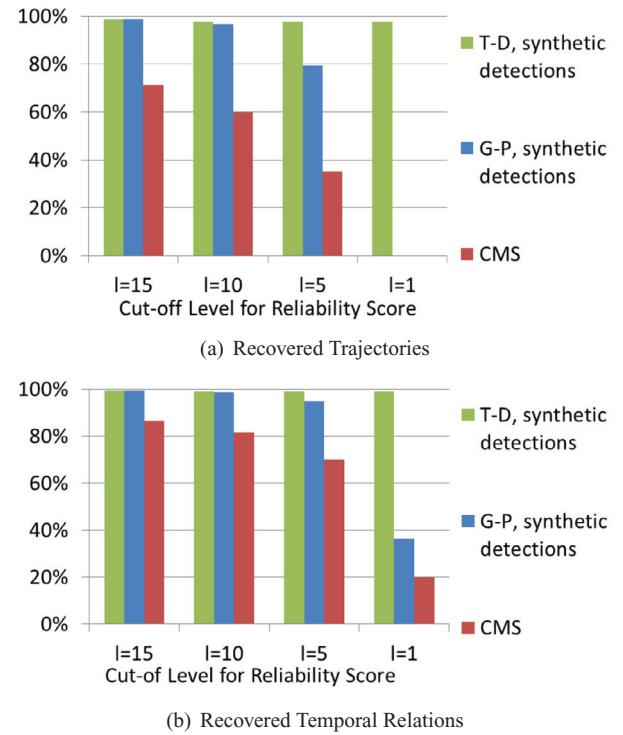


Fig. 6. Results based on synthetic ground truth detections: (a) percentage of trajectories with a reliability score above 95%. (b) percentage of recovered temporal relations – i.e., trajectory points with $\epsilon_f \leq l$ for different cut-off levels l (average of all trajectories together).

neither about synthetic detections nor about the trajectories. For each of the 286 trajectories we calculate the per frame errors ε_f along with their trajectory average ε (see Eq. (5)). We also compute for each of the trajectories the reliability score ς_l for the cut-off levels $l = 1, l = 5, l = 10$ and $l = 15$ (see Eq. (6)). All measures are computed for each of the three sets of output trajectories obtained by T-D, G-P and CMS, respectively.

In Fig. 6 we summarize the result. The percentages of recovered trajectories as well as of recovered temporal relations for T-D, based on the ground truth detections and a cut-off level $l = 1$, illustrate that the graph algorithm is capable to correctly recover the ground truth trajectories. Only in a few cases ($\approx 2\%$), for instance when a point of the trajectory of a neighboring cell is closer to the respective geometrical path, some trajectory points are incorrectly associated.

Fig. 6 also shows the ratios for the remaining cut-off levels. Based on the correct recovery of ground truth tracks we conclude that the per frame errors ε_f of G-P using ground truth detections occur due to the lack of precision of the manually provided geometrical paths P. Only with cut-off levels $l > 10$ G-P is capable to recover all temporal relations ($\approx 99\%$).

For $l = 1$ there are no recovered trajectories at all, neither for G-P with synthetic detections nor for CMS (see Fig. 6(a)).

For $l = 10$ we obtain 99% of recovered temporal relations and 98% of recovered trajectories for G-P (synthetic detections) whereas we still have 21% of missing trajectories for $l = 5$. Based on these results we conclude that the biologists are accepting a level of precision (at least for the present application) corresponding to a cut-off level of $l = 10$.

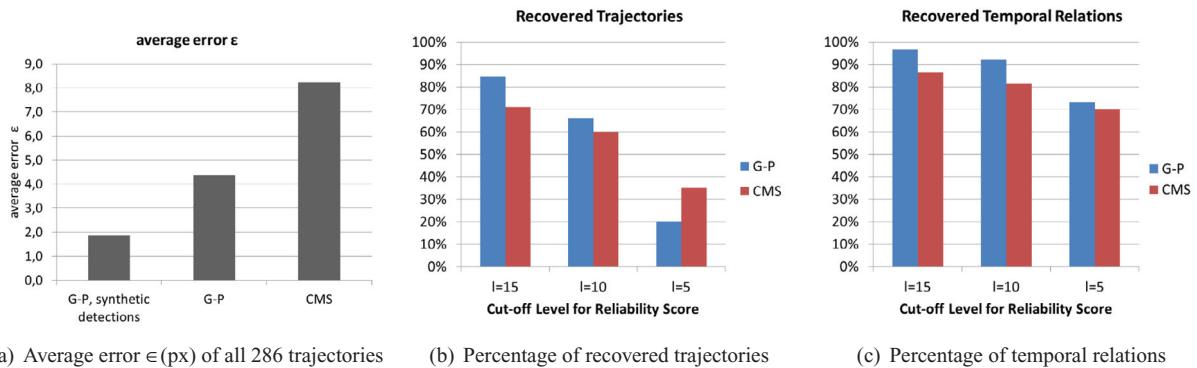
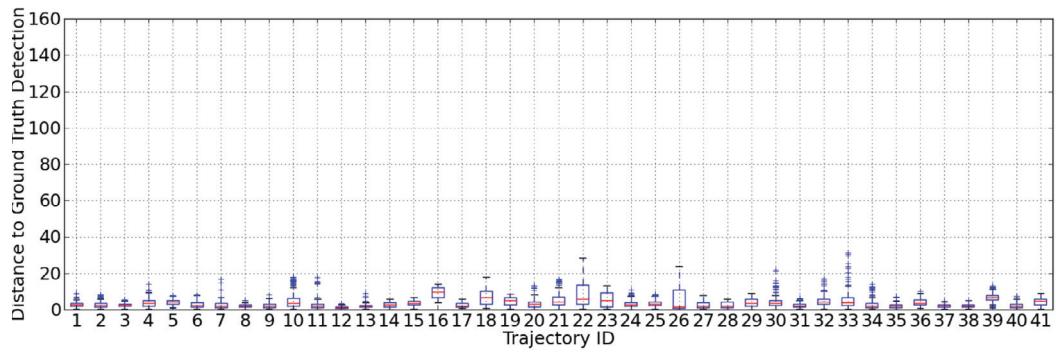
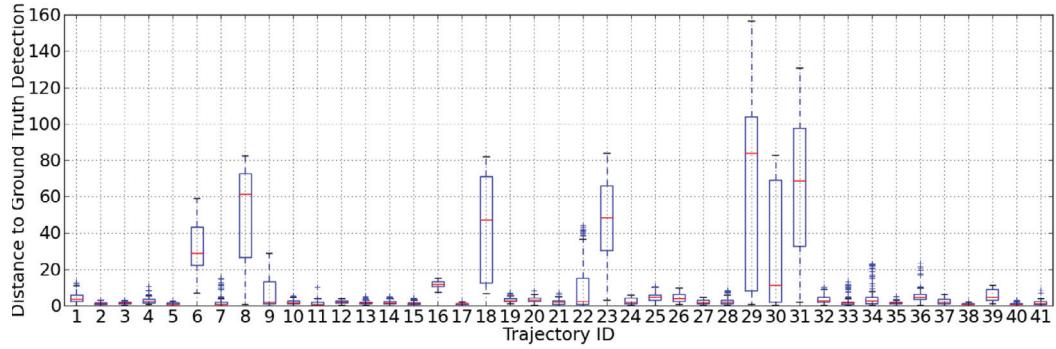


Fig. 7. Comparison of G-P and CMS based on the error measures introduced in Section 3.2.1. The figures compare the average error ε , the ratio of recovered trajectories as well as the ratio of recovered temporal relations for three different cut-off levels l averaged over all 286 cell trajectories.



(a) Per trajectory (cell) errors for G-P



(b) Per trajectory (cell) errors for CMS

Fig. 8. Box plots of the distance to ground truth errors ε_f for the 41 cell trajectories of sequence S_2 . The figures show that complete outliers present in the CMS results (trajectory 6, 8, 18, 23, 29, 30, 31) can be avoided when using our user-path guided spatio-temporal re-parametrization (G-P). The large error boxes of CMS correspond to target lost events in terms of tracking (compare Fig. 9(a) and (c)).

3.4. Experiment 2 – evaluation based on fully automated cell detection

In order to evaluate the performance of G-P under realistic conditions we replace the synthetic detections of Experiment 1 by detections computed automatically using the detector described in Section 3.2.2. As in Experiment 1 we calculate the error measures ε and ε_g for a comparison of G-P and CMS. The resulting performance

values are presented in Fig. 7(a) and visualized in detail for the individual trajectories using box-plots in Fig. 8. The latter are shown for Sequence S_2 , which is representative for all sequences. G-P achieves an average error of 4.4 pixel. The error of CMS in contrast is almost twice as big with an average error above 8 pixels.

Although there exist tracks where CMS outperforms our method in terms of tracking accuracy, Fig. 8(b) shows that there are also

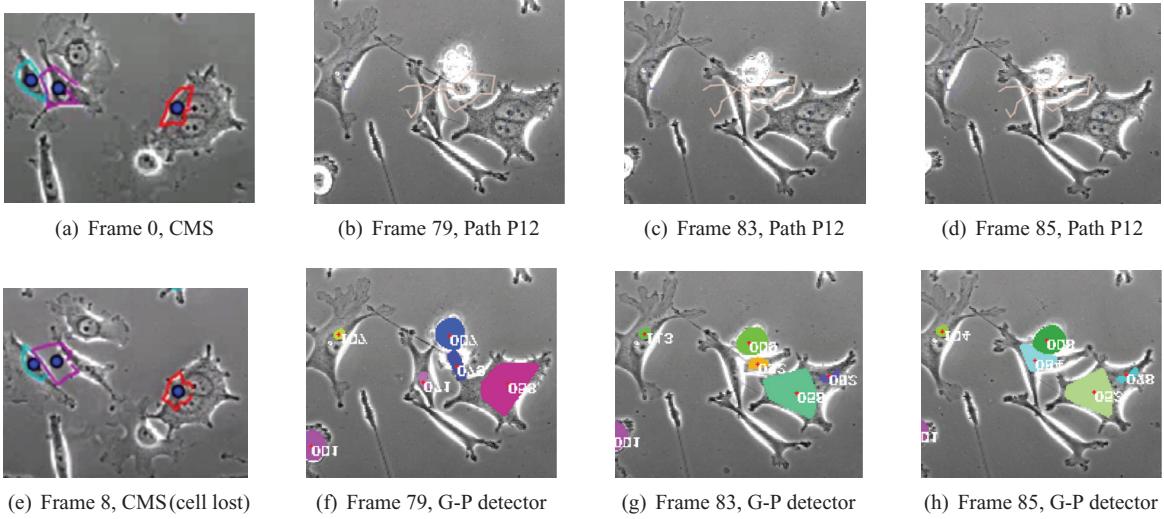


Fig. 9. Result example for trajectory 12 in Sequence S_4 . Figures (a) and (e) show the output of CMS for three cells. In figure (e) we can observe that CMS lost the track of the cell outlined in magenta in frame 8. Figures (b), (c) and (d) show different frames along with the geometrical user path. Figure (f), (g) and (h) show the cell detector output produced by Kazmar et al. (2010). The red dots in the colored regions indicate the detection points used for the proposed re-parametrization.

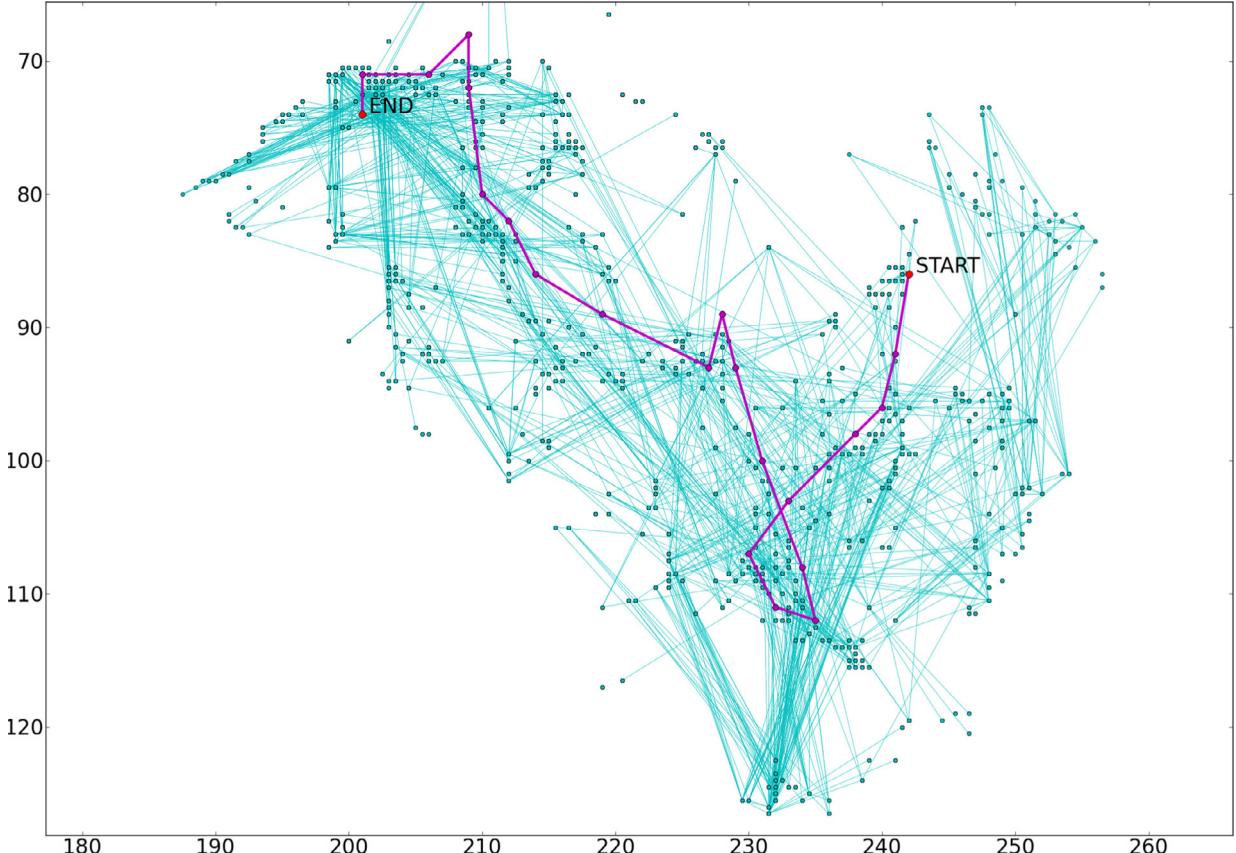
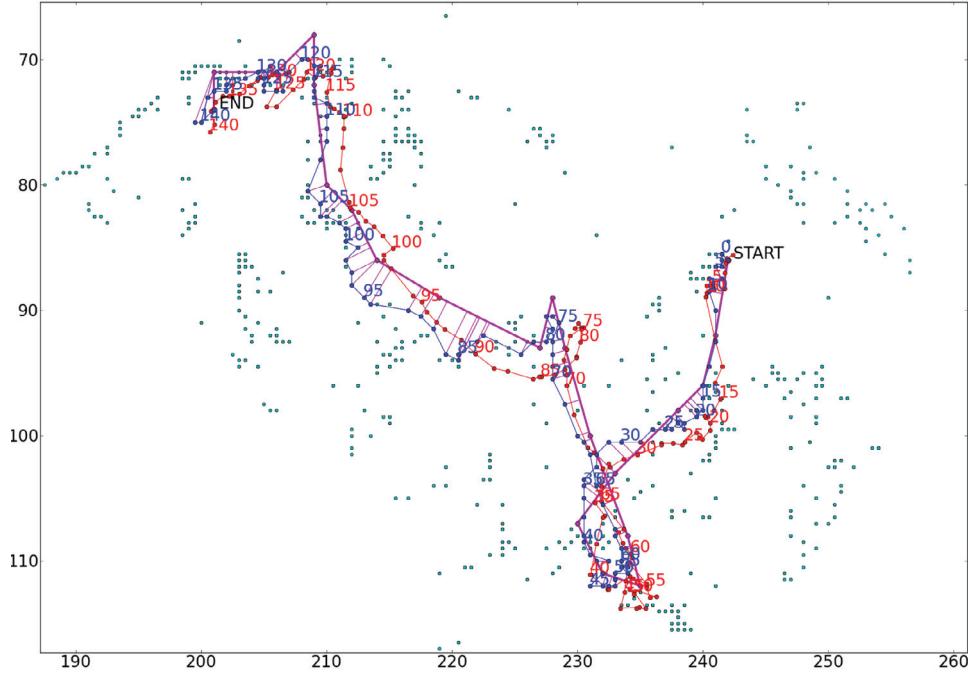


Fig. 10. Projections-graph G for path P1 in Sequence S_2 . The geometric path provided by the user is shown in magenta. The edges in cyan indicate possible transitions in the re-parametrization graph. The edges of the original projections graph G are propagated back to the detections space for the purpose of visualization (compare text in Section 2). ($f_{\text{skip}} = 20$, $g_{D2D}^{\min} = 20$, $g_{D2D}^{\max} = 30$). The labels of the x- and y-axis correspond to the pixel positions in x and y, respectively.

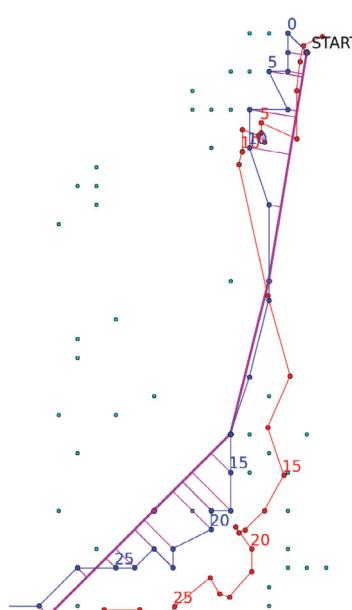
cases where CMS completely loses the cell (cells 6, 8, 18, 23, 29, 30, 31). This is expected since there is no additional user information for CMS helping to get back on the right trajectory once the target cell is lost. Overall CMS fails for 36 of the 286 cell trajectories completely (error $\varepsilon > 15$). In comparison this happens only three times for G-P.

Likewise, we compare the recovered trajectories as well as the recovered temporal relations for the following cut-off values: $l = 5$, $l = 10$ and $l = 15$. The results are shown in Fig. 7(b) and 7(c).

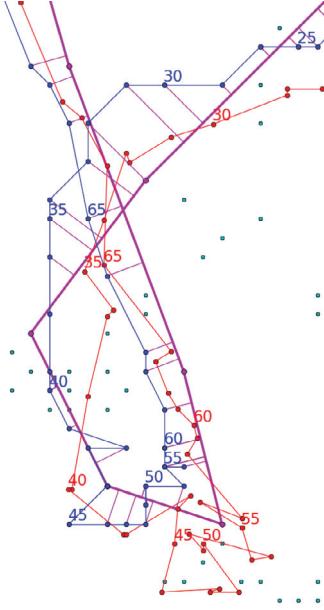
With respect to the level of precision accepted by the biologists derived from Experiment 1 (cut-off level $l = 10$), G-P reaches 92% of recovered temporal relations (compared to 81% achieved by CMS). The



(a) User path (magenta), re-parametrization by G-P (magenta, thin lines), track of detections T-D (blue), ground truth (red). The remaining cyan dots are unused detections of neighboring cells. Along with the ground truth and the path of detections we plot every fifth frame number.



(b) Detail for Sub-figure (a) showing the beginning of the sequence. The frame numbers of the ground truth (red) correspond to the frame numbers of the track of detections (blue).



(c) Detail for Sub-figure (a) showing a cycle in the geometric user path. The figure shows that the method is capable to deal with cyclic structures as well as self intersections in the geometric path (cf., Figure 4(c)).

Fig. 11. Re-parametrization result for trajectory 1 in Sequence S₂ using G-P. The results correspond to the graph G shown in Fig. 10. The figure shows the global result (trajectory) along with two detail views to give a better intuition for the re-parametrization procedure.

percentage of recovered trajectories is 66% for G-P versus 60% for CMS. If we extend the cut-off level to $l = 15$ we achieve 97% of recovered temporal relations for G-P and 86% for CMS and 85% (G-P) versus 71% (CMS) of recovered trajectories.

3.4.1. Illustrative examples

We illustrate by an example a situation where G-P is able to track through an agglomeration of cells whereas CMS is failing. Fig. 9 shows different steps during the attempt of recovering the trajectory corresponding to Path P12 of sequence S_4 . We select this cell as it provides a nice example to address the issues described. Fig. 9(a) depicts for Frame 0 of S_4 the initial CMS-models situated around the cells. Cell C_{12} corresponding to Path P12 is shown in magenta. Since cell C_{12} is close to another cell (shown in cyan) its model already touches this neighboring cell in the first frame. Due to a sudden change in shape of C_{12} (it loses one of its three long extensions from one frame to the next) CMS loses the cell completely in Frame 8 (see Fig. 9(c)).

Fig. 9(b-d and f-h) provides insight into possible difficulties when recovering a trajectory with G-P. Cell C_{12} appears with a circular shape and has a white halo (appearing as a white annulus) in Frame 79 (Fig. 9(b)). It is recognized by our detector in Fig. 9(f) where it is marked by the lower one of the two blue blobs. In the subsequent frames shown in the Fig. 9(c and d) C_{12} changes its shape and is approaching its neighboring cells. The output of our detector is marked by an orange and by a light blue blob in Frames 83 and 85, respectively (Fig. 9(g-h)). As C_{12} is very close to the adjacent cell on its left the detector fails in Frame 85 and marks both cells as a single detection. However, this merged detection is wrong for cell C_{12} and would lead to a notable error in the recovered trajectory. By setting $f_{\text{skip}} > 1$, G-P is able to skip this false detection and succeeds in correctly tracking the cell. f_{skip} is defined depending on the expected number of consecutive frames for which the detector is expected to fail.

To also provide an intuition for the process of graph construction as well as re-parametrization, in Fig. 10, we show a real example for a projection graph G with all nodes and edges computed by G-P. For the purpose of illustration, the edges—corresponding to two projections each—are propagated back into the detections space. For visualization we select path P1 of sequence S_2 since it contains a loop and is therefore illustrative for the method proposed. The geometric user path is plotted as a poly-line overlay in magenta. "START" and "END" indicate the direction of the underlying cell movement. Fig. 11 which corresponds to this graph shows the user path (in magenta) and its re-parametrization along with the associated track of detections. In addition it depicts the ground truth trajectory of the respective cell C_1 . The projections onto the path used for re-parametrization are plotted as thin magenta lines. In Fig. 11(b) and (c) we further provide a detailed view of interesting sections of the respective sequence. The figures allow to follow the track of detections T-D (in blue) from "START" to "END" along the edges of the graph. For each fifth frame the frame number is printed next to a detection or ground truth trajectory extracted from this frame. A comparison with the ground truth (in red) shows that the re-parametrization is correct for the current sequence although there is a loop present in the geometric path. For additional details about the proceeding of G-P in such a situation see Fig. 4(c).

3.4.2. Detection sharing

As already described in Section 1.3 our approach processes the single cells iteratively and additionally allows detection sharing. Therefore, we are interested in the cases where detection sharing actually occurs. When investigating the tracks of detections T-D for all 286 cell trajectories we observe 15 cases of detection sharing between 29 cells (one double sharing). A closer look at the single cells reveals that for only 6 paths ($\approx 2\%$) re-parametrization fails completely having errors > 20 pixel. On the contrary we achieve average errors ε below 5 pixel for 16 of the involved cells.

3.4.3. Run time

On a standard PC (Intel(R) Core(TM) i7 CPU 870, 2.93 GHz, 8 GB RAM, 64-bit OS) we spend several seconds to minutes per path depending on the number of nodes. The number of nodes of course also influences the number of directed edges in the graph. In total we have approximately 10,000 to 21,000 cell detections in each of the sequences out of which the nodes for the respective trajectories are determined by gating ($f_{\text{skip}} = 20$, $g_{\text{D2D}}^{\min} = 20$ and $g_{\text{D2D}}^{\max} = 30$). The number of projection nodes for a trajectory after gating varies between 600 and 6,000 depending on the respective cell density.

4. Conclusion

We formulated the novel problem of recovering spatio-temporal trajectories from (i) approximate geometrical paths without temporal information and (ii) a temporal sequence of sets of noisy detections. Thereby, we imposed a set of constraints on the trajectory to be recovered and we proposed an algorithm based on a graph and shortest path search. The nodes of this graph are selected from the projections of the detections onto the path. The edges between the projections (nodes) are constrained by the temporal order of the projections, by their respective distance on the path and by the spatial distance in between their associated detections. Then, according to the costs associated to the edges, the shortest path in the graph—relating beginning and end of the geometrical path—is determined. The result of this step is a track of projections which is finally used to re-parametrize the geometrical path (algorithm G-P).

As a practical application we investigated in this paper the semi-automated tracking of cancer cells in phase contrast video-microscopy. In order to obtain a reliable tracking result with a user interaction time acceptable for biologists we defined and thoroughly evaluated the following procedure: First, the user draws a geometrical path on an overlay over the video which is playing behind the overlay. Thus, user interaction time is saved compared to manually tracking cells frame by frame. Second, the spatio-temporal associations between frame and position of the corresponding cell on the path in this frame are established automatically.

Our interaction model (G-P) goes along with a higher user interaction time compared to a semi-automated approach (algorithm CMS, see Section 3.2.3) which places seed points in the first frame of the sequence. This additional interaction time of G-P is accompanied by the benefit of a reliable result: In an evaluation on 286 trajectories from seven image sequences we obtained with CMS an average error $\varepsilon = 8$, 2 px which is 86% higher compared to G-P ($\varepsilon = 4$, 4 px). For a precision level accepted by the biologists (10 px) G-P recovers 92% of temporal relations correctly, versus 81% for CMS. CMS fails for 36 trajectories out of 286 completely ($\varepsilon > 15$ px) whereas G-P fails only for 3 trajectories. Compared to manual frame by frame tracking G-P achieves still acceptable precision.

Acknowledgments

This work was mainly carried out inside the CANCERMOTISYS project <http://www.cancermotisys.eu/>, supported by BMWF, Austria, and BMBF, Germany, and partially inside the Austrian COMET program, FFG, Austria.

References

- Al-Kofahi, O., Radke, R.J., Goderie, R.K., Shen, Q., Temple, S., Roysam, B., 2006. Automated cell lineage construction: a rapid method to analyze clonal development established with murine neural progenitor cells. *Cell Cycle* 5 (3), 327–335.
- Bise, R., Yin, Zhaozheng, Kanade, T., 2011. Reliable cell tracking by global data association. In: IEEE International Symposium on Biomedical Imaging: From Nano to Macro, pp. 1004–1010.
- Blackman, S., Popoli, R., 1999. Design and Analysis of Modern Tracking Systems. Artech House Publishers.

- Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.Van, 2009. Robust tracking-by-detection using a detector confidence particle filter. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 1515–1522.
- Carter, Brain C., Shubeita, George T., Gross, Steven P., 2005. Tracking single particles: a user-friendly quantitative evaluation. *Phys. Biol.* 2 (1), 60.
- Catlett, Natalie L., Weisman, Lois S., 2000. Divide and multiply: organelle partitioning in yeast. *Curr. Opin. Cell. Biol.* 12 (4), 509.
- Chenouard, Nicolas, Smal, Ihor, De Chaumont, Fabrice, Maska, Martin, Sbalzarini, F., Ivo, Gon, Yuanhao, Cardinale, Janick, Carthel, Craig, Coraliuppi, Stefano, Winter, Mark, Andrew, R.Cohen, William, J.Godinez, Rohr, Karl, Kalaidzidis, Yannis, Liang, Liang, Duncan, James, Shen, Hongying, Magnusson, Klas, Jaldén, Joakim, Paul-Gilloteaux, Perrine, Roudot, Philippe, Kervrann, Charles, Waharte, François, Tinevez, Jean-Yves, Willemse, Joost, Celler, Katherine, Dan, Han-Wei, Tsai, Yuh-Show, De Solorzano, Carlos Ortiz, Olivo-Marín, Jean-Christophe, Meijering, Erik, 2014. An objective comparison of particle tracking methods. *Nature Methods* 11, 281–289.
- Cordelières, Fabrice P., Petit, Valérie, Kumasaka, Mayuko, Debeir, Olivier, Letort, Véronique, Gallagher, Stuart J., Larue, Lionel, 2013. Automated cell tracking and analysis in phase-contrast videos (itrack4u): development of java software based on combined mean-shift processes. *PLoS One* 8 (11), e81266.
- Costa, Lda Fontoura, Cintra, L.C., Schubert, D., 2005. An integrated approach to the characterization of cell movement. *Cytometry* 68A (2), 92–100.
- Debeir, O., van Ham, P., Kiss, R., Decaestecker, C., 2005. Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes. *IEEE Trans. Med. Imaging* 24 (6), 697–711.
- Decaestecker, C., Debeir, O., Ham, P.Van, Kiss, R., 2007. Can anti-migratory drugs be screened in vitro? a review of 2D and 3D assays for the quantitative analysis of cell migration. *Med. Res. Rev.* 27 (2), 149–176.
- Dijkstra, E.W., 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1 (1), 269–271.
- Jaqaman, Khuloud, Loerke, Dinah, Mettlen, Marcel, Kuwata, Hirotaka, Grinstein, Sergio, Schmid, Sandra L., Danuser, Gaudenz, 2008. Robust single-particle tracking in live-cell time-lapse sequences. *Nat. Methods* 5 (8), 695–702.
- Jiang, Hao, Fels, S., Little, J.J., 2007. A linear programming approach for multiple object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07., pp. 1–8.
- Kan, A., Chakravorty, R., Bailey, J., Leckie, C., Markham, J., Dowling, M.R., 2011. Automated and semi-automated cell tracking: addressing portability challenges. *J. Microsc.* 244 (2), 194–213.
- Kazmar, T., Šmid, M., Fuchs, M., Luber, B., Mattes, J., 2010. Learning cellular texture features in microscopic cancer cell images for automated cell-detection. Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, pp. 49–52.
- Li, Fuhai, Zhou, Xiaobo, Ma, Jinwen, Wong, S.T.C., 2010. Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis. *IEEE Trans. Med. Imaging* 29 (1), 96–105.
- Li, Kang, Chen, Mei, Kanade, Takeo, Miller, Eric, Weiss, Lee, Campbell, Phil, 2008. Cell population tracking and lineage construction with spatiotemporal context. *Med. Image Anal.* 12 (5), 546–566.
- Liu, Anan, Li, Kang, Kanade, Takeo, 2012. A semi-Markov model for mitosis segmentation in time-lapse phase contrast microscopy image sequences of stem cell populations. *IEEE Trans. Med. Imaging* 31 (2), 359–369.
- Magnusson, K., Jaldén, J., Gilbert, P., Blau, H., 2014. Global linking of cell tracks using the viterbi algorithm. *IEEE Trans. Med. Imaging* PP (99), 1.
- Mallik, Roop, Carter, Brian C., Lex, Stephanie A., King, Stephen J., Gross, Steven P., 2004. Cytoplasmic dynein functions as a gear in response to load. *Nat.* 427 (6975), 649.
- Maska, M., Danek, O., Garasa, S., Rouzaut, A., Munoz-Barrutia, A., Ortiz-de Solorzano, C., 2013. Segmentation and shape tracking of whole fluorescent cells based on the Chan-Vese model. *IEEE Trans. Med. Imaging* 32 (6), 995–1006.
- Maška, Martin, Ulman, Vladimír, Svoboda, David, Matula, Pavel, Matula, Petr, Ederra, Cristina, Urbiola, Ainhoa, España, Tomás, Venkatesan, Subramanian, Balak, Deepak M.W., Karas, Pavel, Bolcková, Tereza, Streitová, Markéta, Carthel, Craig, Coraliuppi, Stefano, Harder, Nathalie, Rohr, Karl, Magnusson, Klas E.G., Jaldén, Joakim, Blau, Helen M., Dzyubachyk, Oleh, Pavel, Krízek, Hagen, Guy M., Pastor-Escuredo, David, Jimenez-Carretero, Daniel, Ledesma-Carbayo, María J., Muñoz-Barrutia, Arrate, Meijering, Erik, Kozubek, Michal, Ortiz-de-Solorzano, Carlos, 2014. A benchmark for comparison of cell tracking algorithms. *Bioinform.* 30 (11), 1609–1617.
- Meijering, Erik, Dzyubachyk, Oleh, Smal, Ihor, 2012. Methods for cell and particle tracking. In: Michael conn, P. (Ed.), Imaging and Spectroscopic Analysis of Living Cells Optical and Spectroscopic Techniques. In: Methods in Enzymology, vol. 504. Academic Press, pp. 183–200. (Chapter 9).
- Padfield, D., Rittscher, J., Roysam, B., 2008. Spatio-temporal cell segmentation and tracking for automated screening. In: 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008, pp. 376–379.
- Rahim, M., Bellemare, M.-E., Pirró, N., Bulot, R., 2011. Automatic estimation of pelvic organ anatomical references. In: Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE, pp. 5124–5127.
- Robert, K., 2010. Bringing richer information with reliability to automated traffic monitoring from the fusion of multiple cameras, inductive loops and road maps. *AVSS* 10, 9–16.
- Rohr, Karl, Godinez, William J., Harder, Nathalie, Wörz, Stefan, Mattes, Julian, Tvaruskó, Wolfgang, Eils, Roland, 2010. Tracking and quantitative analysis of dynamic movements of cells and particles. *Cold Spring Harb. Protoc.* (6).
- Sbalzarini, I., Koumoutsakos, P., 2005. Feature point tracking and trajectory analysis for video imaging in cell biology. *J. Struct. Biol.* 151 (2), 182–195.
- Scott, C., 2004. Improved GPS positioning for motor vehicles through map matching. In: Proceedings of 7th International Technical Meeting of the Satellite Division of the Institute of Navigation.
- Snidaro, L., Foresti, G.L., Niu, R., Varshney, P.K., 2004. Sensor fusion for video surveillance. In: Proceedings of the Seventh International Conference on Information Fusion, 2, pp. 739–746.
- Lin, Y.H., Chang, W.L., Hsieh, C.L., 2014. Shot-noise limited localization of single 20 nm gold particles with nanometer spatial precision within microseconds. *Opt. Express* 22 (8), 9159–9170.
- Zhang, Li, Li, Yuan, Nevatia, R., 2008. Global data association for multi-object tracking using network flows. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, pp. 1–8.
- Zimmer, C., Labrugere, E., Meas-Yedid, V., Guillen, N., Olivo-Marín, J.-C., 2002. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. *IEEE. Trans. Med. Imaging.* 21 (10), 1212–1221.