Bioimage informatics

Graphical model for joint segmentation and tracking of multiple dividing cells

Martin Schiegg1,†, Philipp Hanslovsky1,†, Carsten Haubold1, Ullrich Koethe1, Lars Hufnagel2 and Fred A. Hamprecht1,*

1University of Heidelberg, IWR/HCI, 69115 Heidelberg, Germany and 2European Molecular Biology Laboratory (EMBL), Cell Biology and Biophysics Unit, 69117 Heidelberg, Germany

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.
Associate Editor: Jonathan Wren

Received on August 11, 2014; revised on October 14, 2014; accepted on November 12, 2014

Abstract

Motivation: To gain fundamental insight into the development of embryos, biologists seek to understand the fate of each and every embryonic cell. For the generation of cell tracks in embryogenesis, so-called tracking-by-assignment methods are flexible approaches. However, as every two-stage approach, they suffer from irrevocable errors propagated from the first stage to the second stage, here from segmentation to tracking. It is therefore desirable to model segmentation and tracking in a joint holistic assignment framework allowing the two stages to maximally benefit from each other.

Results: We propose a probabilistic graphical model, which both automatically selects the best segments from a time series of oversegmented images/volumes and links them across time. This is realized by introducing intra-frame and inter-frame constraints between conflicting segmentation and tracking hypotheses while at the same time allowing for cell division. We show the efficiency of our algorithm on a challenging 3D+t cell tracking dataset from Drosophila embryogenesis and on a 2D+t dataset of proliferating cells in a dense population with frequent overlaps. On the latter, we achieve results significantly better than state-of-the-art tracking methods.

Availability and implementation: Source code and the 3D+t Drosophila dataset along with our manual annotations will be freely available on http://hci.iwr.uni-heidelberg.de/MIP/Research/tracking/

Contact: fred.hamprecht@iwr.uni-heidelberg.de

Supplementary information: Supplementary material is available at Bioinformatics online.

1 Introduction

Fueled by new microscopic techniques (e.g. Krzic et al., 2012; Tomer et al., 2012), which allow to record in vivo multidimensional images in high spatial and temporal resolution, and by robotic high-throughput setups, biology is developing a great hunger for robust and accurate automated cell tracking (González et al., 2013; Kanade et al., 2011; Maška et al., 2014; Meijering et al., 2009, 2012). As an example, one major goal in developmental biology is the digitization of embryogenesis and its computational analysis, where cell tracking plays an important role. Great advances in this field have been reported most recently (Amat et al., 2014), and one key feature in their study is that they do not strictly separate the cell detection and segmentation stage from the cell tracking stage (For brevity, we mostly refer to the combination of detection and segmentation as detection only). Amat et al. (2014) instead propagate the cell centroids and their approximated Gaussian shape from
the past timesteps to the next while detecting cell divisions at the same time. Despite handling detection and tracking separately, tracking-by-assignment algorithms (Bise et al., 2011; Kausler et al., 2012; Padfield et al., 2011; Schiegg et al., 2013), on the other hand, have proven to be more flexible in terms of modeling power when injecting prior knowledge: biological laws can be modeled as constraints (see Section 3.2) and prior beliefs about individual detections and assignments may be incorporated by utilizing local classifiers trained on a small subset of the data (see Section 3.3) rather than using heuristic rules. Furthermore, tracking-by-assignment models allow for global optimization, which will further improve accuracy, as the assignment problems are solved in a larger temporal context.

Nevertheless, this modeling power in tracking-by-assignment approaches comes at the cost of propagating errors from the first stage (segmentation) to the second (tracking) and insight from the second stage cannot be used to lift ambiguities arising in the first stage. In other words, the tracking result is highly dependent on the detection/segmentation quality, and the overall achievable quality is limited by the lack of interaction between detection and assignment decisions.

Our work aims at solving this particular problem by introducing a method for joint segmentation and tracking in one graphical model. Instead of a single fixed segmentation as used in previous tracking-by-assignment models, the detection phase generates superpixels/voxels from which regions (possible cell segmentations) are extracted as sets of the original superpixels. In particular, these regions can be understood as a selection of possible segmentation hypotheses. Global temporal and spatial information guides the selection of those hypotheses that best fit the overall tracking. During inference, each superpixel is assigned either a cell track identifier or the identifier of the background (cf. Fig. 1). Put another way, our algorithm simultaneously produces both a valid cell segmentation and an assignment of each cell to its cell lineage.

Our main contribution is the formulation of a probabilistic graphical model for joint segmentation and tracking for divisible and almost indistinguishable cells. This undirected graphical model incorporates prior beliefs from multiple local classifiers and guarantees consistency in time and space. We also present a method to generate an oversegmentation, which respects the borders between cells and generates an overcomplete set of superpixels even for cells in dense populations. Furthermore, the 3D+t Drosophila dataset we use for evaluation and our dense manual annotations are provided on our website. This is the first dataset of this size and kind for which manual annotations are freely available.

1.1 Joint detection and tracking

Joint object detection and tracking is handled naturally in tracking algorithms based on active contours (Xiong et al., 2006), space-time segmentation (Lezama et al., 2011) or video segmentation of multiple objects (Budvytis et al., 2011; Vazquez-Reina et al., 2010). However, these methods either cannot deal naturally with divisible objects and heuristics must be used or they cannot cope with dense object populations where objects may overlap. In a very recent study, Amat et al. (2014) present a fast pipeline to simultaneously segment and track cells by propagating Gaussian mixture models through time, but again heuristic rules remain to detect cell divisions. Furthermore, optical flow has been extended to jointly deal with segmentation and tracking (Amat et al., 2013). These authors propose to augment an optical flow algorithm by a regularization term based on similarities of neighboring superpixels modeled in a Markov random field.

In tracking-by-assignment models, however, joint optimization of segmentation and tracking is only rarely tackled. Instead, to reduce errors in the final results, errors are minimized in each step of the two-stage tracking-by-assignment separately, the segmentation step and the tracking step: for the former, specialized segmentation algorithms are used to detect cells at the boundaries between timesteps to guarantee a consistent number of objects contained in each detected region. In a postprocessing step, they correct the original segmentations. Our idea goes one step further and aims to avoid segmentation errors already in the first place by jointly optimizing segmentation (i.e. selection of foreground superpixels) and tracking.

Most similar to our proposed method are the models in Funke et al. (2012), Hofmann et al. (2013) and Jug et al. (2014). Funke et al. (2012) propose an algorithm, which segments an anisotropic 3D volume of branching neurons by generating segmentation hypotheses in 2D slices separately and posing constraints between overlapping segmentation hypotheses. In contrast to our model, the authors do not need to model background for their specific use-case, whereas in our domain, it is important to infer both whether a segment should be activated as foreground and to which segments in the consecutive timesteps it should be linked. Moreover, they do not model detection variables directly but introduce additional transition variables, which model appearance, disappearance and divisions. This is in contrast to our model, where the detection variables allow to model a prior on the count of cells in sets of

Fig. 1. An excerpt of three consecutive timesteps of the Drosophila dataset (2D slices out of 3D volumes). The raw data (top row) is oversegmented into superpixels (middle row). Our graphical model then tracks the cells over time and assigns each segment to a track (indicated by the same random color) or background (black). Offspring cells are assigned the color of their parent cell after mitosis (here: orange). Note that one cell may be represented by multiple superpixels. Scale bars are 10 μm.
regions. The authors in Hofmann et al. (2013) propose a similar idea for joint tracking and object reconstruction from multiple cameras. Both methods have in common that they solve an integer linear program with a large set of hard constraints between superpixels within one (time/z-slice) instance and across instances. In independent work, Jug et al. (2014) jointly segment and track bacteria in 1D+t.

The original idea to refine a segmentation by modeling the conflicts between multiple overlapping segmentation hypotheses was introduced by Brendel and Todorovic (2010) and Ion et al. (2011). Although Brendel et al. propose algorithms to efficiently find the best independent sets in a conflict graph, Ion et al. present a complementary approach to search for maximum cliques in the graph of possible hypotheses (where contradicting tiles are not connected). Their ideas were extended to the temporal domain in Brendel et al. (2011), but they cannot deal with dividing objects. Extending this idea to dividing cells is a much harder problem and the main contribution of our article.

2 Approach

The purpose of this work is to segment and track multiple dividing cells in a tracking-by-assignment framework. To avoid error-propagation from the segmentation to the tracking stage, we propose to jointly segment and track the targets based on an oversegmentation. This process is illustrated in Figure 2: we first run an oversegmentation algorithm on the volumes with overlapping cells to generate multiple segmentation hypotheses. This is followed by the construction of a graphical model for the joint segmentation and tracking. It models competing (intra-frame) relations between the potential cell segmentations, which overlap in space, as well as possible inter-frame hypotheses between regions of adjacent timesteps. In this section, we specify each step of this pipeline consecutively, starting with the oversegmentation step.

3 Methods

3.1 Competing segmentation hypotheses

To make joint segmentation and tracking computationally feasible in tracking-by-assignment approaches, the time series of 2D/3D images/volumes must be coarse grained into superpixels/voxels to reduce the problem space (stages II and III in Fig. 2). Note that the resulting superpixels also afford the extraction of more expressive features at the object rather than the pixel level. To this end, first superpixels are obtained, which are as large as possible but at the same time small enough to respect all cell boundaries. Next, neighboring superpixels are grouped to generate different segmentation hypotheses. Here, we choose to merge the superpixels in a hierarchical fashion. However, the proposed model does not rely on or exploit the resulting tree structure, so any other means of generating complementary but conflicting segmentations could be used.

3.1.1 Oversegmentation

In stage II, the purpose is to obtain an oversegmentation on every image, which is sufficiently fine but as coarse as possible. That is, we prefer single segments (superpixels) for (isolated) objects without ambiguities, whereas multiple (smaller) segments are desired in cases where objects overlap in space. To this end, we propose the following oversegmentation algorithm:

1. Obtain a coarse segmentation, which only distinguishes potential foreground from definite background (high sensitivity and low specificity).
2. Automatically select seeds fulfilling the requirements outlined above.
3. Compute the seeded watershed on the foreground mask.
4. Merge resulting segments hierarchically to potential regions.

Here, the first step may be performed by any segmentation algorithm which can be adjusted in a way that only those pixels are predicted as background where we are sufficiently certain. This step’s output is either a hard segmentation or a probability map of the foreground (soft segmentation). Note that typically, it is not desirable to track the resulting connected components directly, because large clusters of cells may be contained in each connected component. Hence, we continue by splitting these connected components into multiple segments. To this end, the watershed algorithm is applied on the probability map of the potential foreground (the foreground mask is obtained by truncating probabilities below a chosen threshold; we choose 0.5). The seeds for the watershed algorithm

![Fig. 2. First, the raw data are oversegmented in all timesteps separately (stage II). Then, in stage III, segmentation hypotheses are generated by merging adjacent segments into bigger segments (e.g. 2, 3 may be merged into 23). From this structure, a graphical model is constructed (stage IV): overlapping segmentation hypotheses are connected by intra-frame conflicts (black: conflicting segmentation hypotheses; blue: local evidence for the number of cells in one connected component) and inter-timestep transition hypotheses are modeled by binary random variables (yellow nodes) indicating whether the corresponding cell in t has moved to, divided to or is not associated with the corresponding cell in t+1. Note that, for simplicity, only one connected component in only two timesteps is visualized. The proposed factor graph in stage IV, in fact, models all detections and all timesteps in one holistic model at once. Also for simplicity, only a small subset of transition variables is shown. After running inference on this factor graph, the most probable selection of active regions (actual cells) and their transitions between timesteps are found as visualized by the two cells marked in yellow and blue in stage IV.](Image)
are the local maxima of the distance transform on the foreground mask. This gives rise to regularly shaped compact segments.

3.1.2 Region merging
Finally, superpixels are grouped into regions, which form possibly competing cell segmentations (stage III in Fig. 2). These segmentation candidates can be generated in very different ways. For simplicity, we choose a hierarchical region merging in a region adjacency graph using $L$ tree levels. Its edge weights between neighboring segments/regions may be arbitrarily complex, and the regions may be merged in an order determined by these edge weights.

Because the segmentation hypotheses are composed from the same superpixels, natural conflicts between these regions exist and are resolved by our graphical model (stage IV in Fig. 2) as discussed in the next section.

3.2 Graphical model for joint segmentation and tracking

3.2.1 Overview
Based on the oversegmentation described in Section 3.1, a graphical model [here: a factor graph (Kschischang et al., 2001)] is constructed whose factors collect evidence from local classifiers and, at the same time, guarantee consistency due to linear constraints. That is, impossible configurations are disallowed, e.g., a cell dividing into more than two children. Building the graphical model corresponds to stage IV in Figure 2. The construction of the factor graph and the meaning of contained factors and random variables are described in detail in this section. We will refer to the toy example depicted in Figure 2 as a running example.

3.2.2 Random variables
To build the factor graph for joint segmentation and trackings, we first introduce two types of binary random variables, detection variables and transition variables. In particular, each possible cell segmentation (region) gets assigned a detection variable $X_{i,t} \in \{0, 1\}$, where $i$ is the connected component containing the region, $a$ is the identifier of the region and $t$ is the timestep. Secondly, variables $Y_{i,t} \in \{0, 1\}$ for each possible inter-frame transition between two regions in adjacent timesteps are added. In our illustrative example in Figure 2, one detection variable is $X_{44,4(44)}$, referring to region 4 in the connected component formed by regions 4 and 5 at time $t + 1$. $Y_{123(23,45)}$ is an exemplary inter-frame transition variable, where the indices mean that region 23 in connected component 123 at time $t$ may be associated with region 4 in connected component 45 at time $t + 1$.

3.2.3 Factors
We continue the construction of our graphical model by adding factors. Factors may disallow specific configurations (see paragraph constraints) and score possible configurations of their associated variables based on estimated posterior probabilities $P$ that are here determined by probabilistic classifiers using local evidence $f_{i,t}$. In the following, intra-frame factors (detection and count factors) and inter-frame factors (outgoing and incoming factors) are described.

Obviously, all regions in each path from a leaf node to the root node in the region merging graph (see stage III of Fig. 2) form competing segmentation hypotheses and are represented by a conflict set $C_k$ each of which contains indices of such conflicting regions. For each such conflict set $C_k$, a higher order detection factor $\psi_{det}$ is added in the graphical model with the energy.

\[
E_{\text{det}}(X_{i} \in F_k) = \begin{cases} 
-\omega_{\text{det}} \log \left( \hat{P}_{f_{i}}(X_{i} = 1) \right), & X_{i} = 1 \\
-\omega_{\text{det}} \max_{X_{i} \in C_k} \log \left( \hat{P}_{f_{i}}(X_{i} = 0) \right) + \epsilon_{\text{bias}}, & X_{i} = 0 \forall X_{i} \in C_k 
\end{cases}
\]

(1)

where $X_{i} = \{X_{i} \in \mathcal{C}_{k}\}$ and $F_0 = \{f_{i} \in \mathcal{C}_{k}\}$ are the detection variables (and their corresponding features) of regions contained in conflict set $C_k$ and $w_{\text{det}}$ weights the detection factor against other factors. A factor $\psi(X)$ can be obtained from the given energy $E(X)$ by the following transformation: $\psi(X) = \exp (-E(X))$. For the sake of brevity, we will only describe the energies in the remainder of the article.

Equation (1) translates to the following: a prior probability $P_{f_{i}}(X_{i} = 1)$ obtained from a pre-trained local classifier (see Section 3.3 for details) with features $f_{i}$ is transformed into an energy for the configuration where exactly one $X_{i} = 1$ is found to be true cell. In the second case, none of the regions in the conflict set is a true cell, a penalty has to be paid based on the classifier’s belief of each of the regions being false-positive detections. The model parameter $\epsilon_{\text{bias}}$ can put a bias on regions to be activated rather than deactivated in case of doubt. Note that impossible configurations, such as the selection of more than one competing region, are forbidden by constraint $\mathcal{C}$, see Section 3.2.4.

In Figure 2, the potential $\psi_{\text{det}}$ ideally obtains a high energy (i.e., low probability) for the single region 2, whereas region 23 has a low energy as it better represents an entire cell.

Moreover, to further leverage local evidence, a higher-order count factor

\[
E_{\text{count}}(\{X_{i}\}) = -w_{\text{count}} \log \left( \hat{P}_{\text{count}} \left( \sum_{X_{i} \in \{X_{i}\}} X = k \right) \right),
\]

(2)

where $\{X_{i}\}$ denotes the detection variables for all regions belonging to connected component $i$ at time $t$. It injects prior beliefs for each connected component $i$ to contain $k$ actual cells. To this end, a probabilistic count classifier (see Section 3.3) is trained using features such as total intensity or size and applied on connected components. For instance, two active regions are favored for connected component 123.

The factors above are both associated with variables from single timesteps only. To achieve temporal associations of cells across timesteps, the model has to be extended by inter-frame factors, which connect detection with transition variables. Firstly, outgoing factors with energy

\[
E_{\text{out}}(X_{i,t}, X_{i,t+1}) = E_{\text{det}}(X_{i,t}; X_{i,t+1}) + E_{\text{move}}(X_{i,t}; X_{i,t+1})
\]

(3)

associate each variable $X_{i,t}$ with all possible transitions $Y_{i,t+1}$ to variables in the successive timestep. This factor is decomposed into three energy terms: disappearance (penalizing the termination of a track), cell division (allowing for cell division, based on estimated division probabilities by a local division classifier) and cell migration (simple association between two cells of consecutive timesteps, based on a local transition classifier).

The second inter-frame factor, the incoming factor, assigns a cost in case a cell appears, i.e., $X_{i,t+1}$ is one, but all of the transition variables in $Y_{i,t+1}$ are zero. Details for the inter-frame factors are provided in the Supplementary Material.

Omitted in these factors so far are impossible configurations, such as more than one ancestor or more than two descendants for one cell. These configurations are prohibited by adding the following constraints.
tors (black squares) connect inter-frame transition with detection variables. The constraints are summarized in Table 1 and explained in the following.

Because overlapping—and hence conflicting—regions are contained in the segmentation hypotheses, constraints need to restrict the space of feasible solutions to non-conflicting solutions. For this purpose, conflicting hypotheses are subsumed into conflict sets \( C_{\text{conf}} \). (Red factors and their associated detection variables in Fig. 3. ) Constraint \( C_{\text{conf}} \) in Table 1 ensures that at most one detection variable is active in each conflict set. Taking conflict set \( C = \{123, 23, 3\} \) may be active at a time. Outgoing and incoming factors (black squares) connect inter-frame transition with detection variables and ensure a unique lineage of cells.

3.2.4 Constraints

We add linear constraints to guarantee that only feasible configurations are part of a solution. Constraints within individual timesteps will be referred to as intra-frame constraints, whereas inter-frame constraints regularize the interaction of detection with transition variables. The constraints are summarized in Table 1 and explained in the following.

3.2.5 Inference

In our global graphical model, the total energy is the sum of all factors over all possible variable configurations of detection variables \( X \) and transition variables \( Y \). It should be noted that \( X \) and \( Y \) contain all random variables of all timesteps taking all information available into account in one holistic graphical model. The probability for a configuration \( X, Y \) is then given by the Gibbs distribution \( P(X, Y) = \lambda^{E(X,Y)} \) and the optimal tracking corresponds to its MAP solution. We solve the energy minimization problem to global optimality by solving the corresponding integer linear program.

After inference, the optimal configuration of the factor graph can be interpreted as a segmentation and tracking result as illustrated in stage IV in Figure 2. The graphical model assigns a track identifier to each foreground superpixel and sets segment values to zero, which are inferred to be background.

3.3 Local classifiers

The factors of the graphical model introduced in Section 3.2 are based on the predictions of local classifiers for

1. the number of cells in a connected component: the count classifier is trained based on the appearance (e.g. the size, intensity and radius) of a connected component and predicts the number of cells that are contained within. The predictions are then injected into the count factors in Equation (2) as prior belief for the number of cells contained in a connected component.
2. true detections: the detection classifier estimates how strongly a region resembles a cell [cf. Equation (1)].
3. cell divisions: the division classifier rates the probability of triples of regions, ancestor and two children from consecutive frames, to represent a division.
4. cell migration (moves): the move classifier rates every pair of regions associated with a transition variable.

In our implementation, we train random forest classifiers, but any classifier which provides (pseudo-)probabilistic predictions can be used. These classifiers are trained on user annotated training examples. We refer the reader to the Supplementary Material for detailed specifications and features used.

3.4 Implementation details

In this cell tracking application, we use the following methods and parameters for the oversegmentation algorithm sketched in Section 3.1. To obtain a coarse foreground mask, we use the segmentation toolkit ilastik (Sommer et al., 2011), which can segment both the phase-contrast images from the Rat stem cells dataset and the stained cell nuclei from the Drosophila dataset: here, prediction maps for each timestep are computed independently using a pixel-wise random forest trained on few training examples from the respective dataset. We use 100 trees in every experiment and select the following features at different scales: Gaussian smoothing, Gaussian Gradient Magnitude, Difference of Gaussians, Structure Tensor Eigenvalues and Hessian of Gaussian Eigenvalues. Then, the seeds are determined by the local maxima of the distance transform on the slightly smoothed foreground mask (Gaussian smoothing with \( \sigma = 0.3 \) and \( \sigma = 1.0 \) in the case of Drosophila and Rat stem cells, respectively) and nearby seeds are pruned by dilating with a disc/ball of radius 2 pixels. Resulting segments are merged hierarchically with edge weights determined by the ratio of the length of their common border and the perimeter of the smaller region. Although much more expressive weights could be used here, we find that these simple features already perform well. Then, at every level \( l \in \{0, ..., L\} \) of the hierarchical segmentation hypotheses (we choose...
the tree depth $L=4$ in the 2D+t and $L=5$ in the 3D+t dataset), edge weights are ordered and the neighbors with the $p\%$ highest weights are merged iteratively. In this way, segments completely contained within other segments are merged first, whereas regions which only touch in few pixels are merged last. Here, we set $p=20$ for $t \in \{0, \ldots, L-1\}$ and $p=100$ for $t=L$ to get the connected components of the foreground mask as the root node of the segmentation hypotheses trees. Our model and implementation is not limited to hierarchical segmentation hypotheses. In fact, any algorithm that generates competing segmentation hypotheses could be used.

The graphical model described in Section 3.2 is implemented in C++ using the open-source library OpenGM (Andres et al., 2012). For tractability, the number of inter-frame hypotheses is pruned to a reasonable number of candidates in the spatial proximity of each region: in particular, inter-frame hypotheses between frames $t$ and $t+1$ are generated by finding the two nearest neighbors in $t+1$ for each region in frame $t$ and the two nearest neighbors in $t$ for each region in frame $t+1$. This procedure yields many inter-frame hypotheses ($>2$) in dense cell populations and only few hypotheses in the parts of the image where cells are sparse. To create training examples for the classifiers, a small subset of the raw data is selected and sparsely annotated to train a random forest (Breiman, 2001) for each classifier suggested in Section 3.3. We choose 100 trees for each and train the random forests to purity. The parameters of the factor graph are then tuned to best fit a small, fully annotated subset of the data. These parameters are used for the final predictions on the entire dataset to report the performance measures. To do inference on our graphical model, we use the (integer) linear programming solver CPLEX. The globally optimal solution for the entire time sequence is found within $\approx 10-70$ min. We refer the reader to the Supplementary Material, Section 5, for a more detailed runtime discussion.

### 4 Results and discussion

We perform comparative experiments on two datasets—a cell culture (2D+t) and a developing *Drosophila* embryo (3D+t dataset). The former is challenging due to severe mutual overlap, whereas the latter is difficult owing to its ambiguity in the segmentation hypotheses due to high cell density under low contrast.

The first dataset is publicly available from Rapoport et al. (2011) (their dataset A) and consists of a time series of 209 images (1376 × 1038 pixels) of about 240 000 pancreatic stem cells of a *rattus norvegicus* (`Rat stem cells`). This dataset is particularly challenging due to the cells changing their appearance (shape, size and intensity) over time from long elongated to round cells. Moreover, the proliferating stem cells quickly grow to a dense population causing frequent overlaps between cells. Because of the dataset’s high temporal resolution, it is difficult to pinpoint a cell division to a specific point in time. Instead, mitosis occurs over multiple timesteps.

For this reason, we subsample the sequence in time, processing every second image only (leaving us with 104 timesteps) and relax the evaluation criterion for divisions (see Section 4.1). We further resample the ground truth provided by (Rapoport et al., 2011) to guarantee that no cell division is lost in the subsampling.

The second dataset is a developing *Drosophila* embryo (Schiegg et al., 2013) (their dataset B). On average, about 800 cells are tracked over about 100 timesteps (730 × 320 × 30 voxels, voxel resolution 0.5 μm). Schiegg et al. (2013) evaluate their tracking method on this dataset conditioned on a given segmentation. To evaluate the performance of our joint approach of segmentation and tracking, we extend their manual annotations such that it also covers previously missing cells and that voxels of falsely merged cells are assigned to individual cell identities (Both the dataset and our manual annotations will be made freely available.). In this way, we can further report segmentation/detection measures in addition to tracking measures unconditioned on the segmentation result.

#### 4.1 Evaluation measures

In contrast to the typical evaluation of tracking-by-assignment methods, for which an evaluation conditioned on the segmentation is sufficient to determine the efficiency of the tracking algorithm, here, both segmentation and tracking must be compared against a ground truth. To evaluate the segmentation quality, we use the Jaccard index as a similarity measure between a region $r_{gt}$ of the result and ground truth region $r_{gt}$, i.e. $\text{Jaccard measure} = m(\cap(r_{gt}, r_{gt})) / m(\cup(r_{gt}, r_{gt}))$. The best-matching region $r_{match}(r_{gt}) = \arg\max_{r_{gt}} \text{Jaccard measure}(r_{gt}, r_{gt})$ for some ground truth region $r_{gt}$ counts as a true-positive segmentation for that region if its Jaccard index is greater than some threshold $\tau$ (we set $\tau = 0.5$) [For (Amat et al., 2014), we choose $\tau = 0.0$ and use a dilated centroid as segment. See Supplementary Material for details.]. Unmatched ground truth/tracking result regions are considered false-negative/false-positive detections.

We then compare the frame-to-frame tracking events (*moves* and *divisions*) from the ground truth to those of the tracking result. We report an *unconditioned* tracking result and *conditioned* performance measures. The former evaluates the tracking on the raw data directly, the latter is conditioned on the true segmentation hypotheses. Note that it is often not clear from the raw data, in which exact timestep a cell division is occurring. We hence allow cell divisions to be off from the ground truth by one timestep, i.e. a division is still counted as a true positive if it occurs one timestep earlier or later within the same track. Finally, based on the number of true/false positives and false negatives, *precision, recall* and *f-measure* are computed for detections, moves and divisions.

#### 4.2 Results for joint segmentation and tracking

To evaluate the performance of our model for joint cell segmentation and tracking, we perform experiments on the two datasets...
described above. We compare with two recently proposed cell-tracking algorithms:

1. A graphical model for cell tracking (Schiegg et al., 2013) (based on a given segmentation), which can correct for falsely merged cells in a post-processing step. To show that our method operates on a reasonably fine oversegmentation and that it is not enough to merely track the superpixels in this oversegmentation, we also perform experiments using the method of (Schiegg et al., 2013) but use our oversegmentation as input. To this end, we set their parameter of maximally allowed cells in a single detection to 1. In all three methods, we use the same count and division classifier, to which in our method move and detection classifiers are added.

2. A cell tracking pipeline designed to track entire embryos (Amat et al., 2014). We evaluate their algorithm on both the raw data directly and our prediction maps as input. Note that this code was made for 3D+t datasets; we refer to our Supplementary Material for further details.

In the 2D+t dataset, we furthermore compare with the results of Rapoport et al. (2011) for the quantitative results reported there.

### 4.2.1 Segmentation quality

We first investigate the quality of cell segmentations, see Table 2 for results. Note that in both ours and Schiegg et al. (2013), cell candidates may be set inactive by the graphical model. In both datasets, our method outperforms the segmentation quality of Schiegg et al. (2013) with an \( f \) measure of 0.97 and 0.93 compared with 0.88 and 0.87. Because our model groups superpixels into cells or deactivates them, it is not crucial in our approach whether cell candidates (or superpixels) are touching in the segmented image. In the method of Schiegg et al. (2013), in contrast, the complexity of the model is determined by the worst case cluster size, i.e. the number of potentially merged cells. Hence, in their approach, the need for correctly segmented individual cells leads to parameter settings that in turn make for many false negatives in the segmentation. We consider it a strong advantage of our method to deal with competing segmentation hypotheses rather than repairing a fixed segmentation. Moreover, Rapoport et al. (2011) achieve on the Rat stems cells data a recall of 0.95 (they do not report precision), whereas our method obtains a recall of 0.96 under very high precision (0.99). Note that Rapoport et al. (2011) use \( t = 0.3 \) (cf. Section 4.1), whereas we set \( t = 0.5 \) as a stronger criterion. Amat et al. (2014) achieve similar or slightly better detection accuracies on the 3D+t dataset, because their parametric model for cell appearance is seemingly a good fit for the 3D+t dataset. Our nonparametric model, in contrast, fares better on the more irregular cell shapes in the 2D+t data, where the detection accuracy of (Amat et al., 2014) only increases in the course of the movie, seemingly due to the following reasons: the cells adopt a Gaussian shape only after a number of frames and their model is tailored toward Gaussian shaped objects. Moreover, because of non-homogeneous illumination, initialization with the correct number of cells seems to be imperfect. Of course, these detection errors in this dataset are also mirrored when inspecting their tracking quality.

### 4.2.2 Tracking quality

The detection/segmentation errors usually propagate to the next stage, the tracking stage. Our model aims at avoiding such error propagation, the performance measures for the tracking quality are reported in Table 3. On both datasets, the proposed method is on par with Schiegg et al. (2013) and Amat et al. (2014) in terms of (frame-to-frame) move events. For the division events, we show through the \( f \) measures of 0.70 (unconditioned) and 0.84 (conditioned) that our method can deal with mitosis in the challenging 2D+t dataset slightly better than Rapoport et al. (2011) (\( f \) measure of 0.67) and improves significantly upon (Schiegg et al., 2013) (\( f \) measures of 0.32 and 0.56, respectively), although using the same classifier. On the other hand, the competitive method (Schiegg et al., 2013) yields a slightly better detection rate of division events on the 3D+t dataset. We believe that this fluctuation is due to a lack of training data for the graphical model (only 16 divisions occur in our training set), which is more critical in our approach because it has more degrees of freedom. In particular, when dealing with oversegmented objects, a strong division classifier is crucial because the introduced ambiguity may lead to increased confusion in division events. If higher division accuracies are desired, the training set needs to be expanded at the cost of more user annotations. Furthermore, the division detection accuracy our proposed model achieves is significantly better than that of (Amat et al., 2014). We believe this is due to the reason that divisions are handled naturally in tracking-by-assignment approaches (compared with heuristic rules), and further evidence can be injected through local classifiers trained on this specific event.

Qualitative results for the 2D+t dataset are presented in the Supplementary Material.

### 5 Conclusion

This work is motivated by the desire to overcome the propagation of errors from a separate segmentation phase to an independent tracking phase in a tracking-by-assignment framework. In response, we propose an undirected graphical model that couples decisions over all of space and all of time. This model simultaneously selects a

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Segmentation</th>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>( f ) measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rat stem cells (2D+t)</td>
<td></td>
<td>Rapoport et al. (2011)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schiegg et al. (2013)</td>
<td>0.75</td>
<td>0.99</td>
<td>0.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schiegg et al. (2013)</td>
<td>0.79</td>
<td>0.99</td>
<td>0.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amat et al. (2014) on raw data</td>
<td>0.94</td>
<td>0.95</td>
<td>0.94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amat et al. (2014) on our prediction maps</td>
<td>0.92</td>
<td>0.95</td>
<td>0.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>0.99</td>
<td>0.96</td>
<td>0.97</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drosophila embryo (3D+t)</td>
<td></td>
<td>Schiegg et al. (2013)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schiegg et al. (2013)</td>
<td>0.82</td>
<td>0.93</td>
<td>0.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schiegg et al. (2013)</td>
<td>0.77</td>
<td>0.95</td>
<td>0.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amat et al. (2014) on raw data</td>
<td>0.97</td>
<td>0.93</td>
<td>0.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amat et al. (2014) on our prediction maps</td>
<td>0.96</td>
<td>0.89</td>
<td>0.93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>0.99</td>
<td>0.88</td>
<td>0.93</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bold values represents best performance results.
Table 3. Quantitative results for cell tracking. Reported are precision, recall and \( f \) measure for (frame-to-frame) events move (i.e. transition assignments) and cell divisions (i.e. mitosis). Rat stem cells comprises 119 266 and 1998 such events, respectively, whereas Drosophila embryo includes 63 548 moves and 226 divisions. Results are shown for the tracking being conditioned on segmentation result and directly compared with ground truth (unconditioned).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Unconditioned</th>
<th>Conditioned on segmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Moves</td>
<td>Divisions</td>
</tr>
<tr>
<td></td>
<td>Prec.  Rec.  ( f ) measure</td>
<td>Prec.  Rec.  ( f ) measure</td>
</tr>
<tr>
<td>Rat stem cells (2D+t) (Rapoport et al., 2011)</td>
<td>0.96 0.89 0.92</td>
<td>0.55 0.87 0.67</td>
</tr>
<tr>
<td>Schiegg et al. (2013) with their segmentation</td>
<td>0.68 0.26 0.32</td>
<td>0.91 0.91 0.95</td>
</tr>
<tr>
<td>Schiegg et al. (2013) with our oversegmentation</td>
<td>0.22 0.44 0.29</td>
<td>0.96 0.68 0.80</td>
</tr>
<tr>
<td>Amat et al. (2014) on raw data</td>
<td>0.62 0.17 0.26</td>
<td>0.97 0.94 0.95</td>
</tr>
<tr>
<td>Amat et al. (2014) on our prediction maps</td>
<td>0.74 0.31 0.44</td>
<td>0.98 0.97 0.98</td>
</tr>
<tr>
<td>Ours</td>
<td>0.97 0.93 0.95</td>
<td>0.74 0.67 0.70</td>
</tr>
<tr>
<td>Drosophila embryo (3D+t) (Schiegg et al., 2013)</td>
<td>0.95 0.85 0.90</td>
<td>0.65 0.74 0.69</td>
</tr>
<tr>
<td>Schiegg et al. (2013) with their segmentation</td>
<td>0.73 0.77 0.75</td>
<td>0.04 0.78 0.08</td>
</tr>
<tr>
<td>Schiegg et al. (2013) with our oversegmentation</td>
<td>0.73 0.77 0.75</td>
<td>0.25 0.75 0.38</td>
</tr>
<tr>
<td>Amat et al. (2014) on raw data</td>
<td>0.91 0.86 0.89</td>
<td>0.18 0.70 0.29</td>
</tr>
<tr>
<td>Amat et al. (2014) on our prediction maps</td>
<td>0.91 0.86 0.91</td>
<td>0.54 0.75 0.63</td>
</tr>
<tr>
<td>Ours</td>
<td>0.96 0.86 0.91</td>
<td>0.54 0.75 0.63</td>
</tr>
</tbody>
</table>

Bold values represent best performance results.

There are several immediately relevant avenues for future work, including structured learning of the classifiers or speed-ups in runtime. The latter may be achieved by domain decomposition, which needs to guarantee consistency in overlaps. relaxations such as dual time. The latter may be achieved by domain decomposition, which needs to guarantee consistency in overlaps. Also approximate solvers may be used to speed up inference. Furthermore, coupling the method of Amat et al. (2014) with our approach might yield significant speed-ups and high accuracy in terms of cell division detection.

Acknowledgement

We thank Christoph Klein (University of Heidelberg) for his assistance in manual tracking annotations.

Funding

This work was partially supported by the Heidelberg University Cluster of Excellence Cell Networks [grant number EXC81] and the HGS Mathcomp [DFG GSC 220].

References