A Discrete Chain Graph Model for 3d+t Cell Tracking with High Misdetection Robustness

Bernhard X. Kausler[1], Martin Schiegg[1], Bjoern Andres[1,2],
Martin Lindner[1], Ullrich Koethe[1], Heike Leitte[1],
Jochen Wittbrodt[3], Lars Hufnagel[4], and Fred A. Hamprecht[1]

[1]HCI/IWR, Heidelberg University, Germany
[2]SEAS, Harvard University, United States
[3]COS, Heidelberg University, Germany
[4]European Molecular Biology Laboratory (EMBL), Heidelberg, Germany
Contact: fred.hamprecht@iwr.uni-heidelberg.de

Abstract. Tracking by assignment is well suited for tracking a varying number of divisible cells, but suffers from false positive detections. We reformulate tracking by assignment as a chain graph–a mixed directed-undirected probabilistic graphical model–and obtain a tracking simultaneously over all time steps from the maximum a-posteriori configuration. The model is evaluated on two challenging four-dimensional data sets from developmental biology. Compared to previous work, we obtain improved tracks due to an increased robustness against false positive detections and the incorporation of temporal domain knowledge.

Keywords: chain graph, graphical model, cell tracking

1 Introduction

One grand challenge of developmental biology is to find the lineage tree of all cells in a growing organism, i.e. the complete ancestry of each cell [17]. The challenges encountered include (i) simultaneously tracking an *unknown* and *variable* number of cells; (ii) allowing for cell *division*; and (iii) high accuracy, because each tracking error affects a complete subtree of the lineage.

We propose a tracking by assignment solution contributing these novelties:

1. In recent years graphical models proved to be exceptionally successful in computer vision [4]. We adopted the approach and present the first probabilistic graphical model for cell tracking which can track an unknown number of divisible objects that may appear or disappear and which can cope with false detections.
2. The model achieves robustness against noise detections by solving the tracking problem for all time steps at once taking the expected time between divisions into account. The most likely cell lineage is obtained as the model configuration with maximum a-posteriori (MAP) probability.

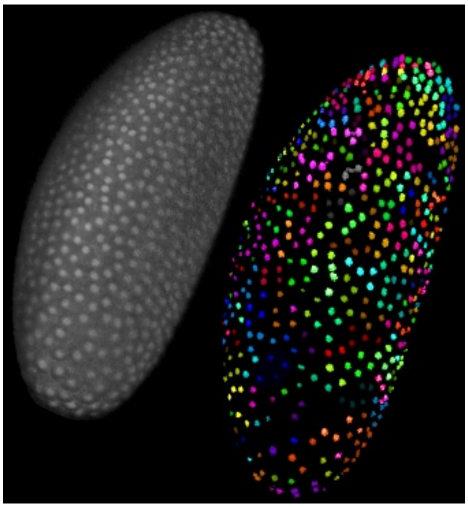


Fig. 1: Maximum intensity projection of a Drosophila embryo and tracking: common ancestry is indicated by common color.

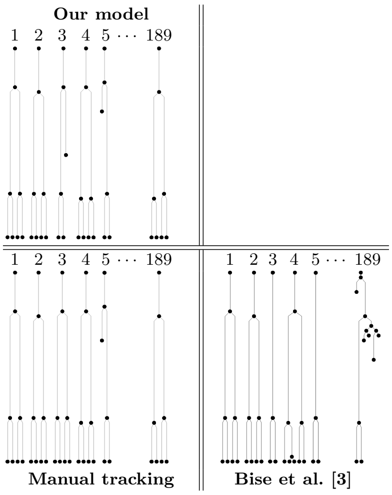


Fig. 2: **Lineage trees for Drosophila**: exemplary manually tracked lineage trees are compared to the results of Bise et al. [3] and our model.

3. While all literature with similar data focuses on single organisms, we present results on zebrafish and–for the first time–on Drosophila together with gold standard lineages and benchmarking tools (as a supplementary).

We obtain the exact MAP solution by integer linear programming (ILP), and–when comparing to the gold standard lineages–obtain results that are superior to existing methods.

2 Related Work

Tracking in science comprises many different application domains and approaches [23]. Our work is concerned with the tracking of a varying number of divisible objects with similar appearance in images with its main application in cell lineage reconstruction. It has to be distinguished from tracking many similar particles that are not dividing [21].

When the temporal resolution of the raw data is high compared to the rate of change, derivative-based methods such as optical flow [18] or level sets [20] can be used to simultaneously segment and track the evolution of one or more targets in spacetime.

A lower temporal resolution–as it is the case in our application–makes the problem harder and in response, most algorithms then separate the detection / segmentation from the tracking problem: detections are found in each time step and fed into a tracking routine. Two different kinds of models are typically used: state space models and assignment models. The Kalman filter, a linear Gaussian state space model, is the archetype of the former class, which interprets detections as caused by a hidden target. It has been generalized in a number of ways allowing for nonlinear motion models, discrete state spaces, non-Gaussian distributions [2, 7], multiple target hypotheses, or even an unknown (but fixed) number of targets [8]. While state space models easily accommodate target properties such as velocity, size, and appearance and are robust against noisy detections by design, they are unfortunately hard to coax into dealing with a variable number of dividing objects.

Tracking by assignment, on the other hand, treats every detection as a potential target itself. It easily accommodates multiple or dividing objects; however, this increased flexibility must be reined in by enforcing the consistency (see section 3) of a tracking. This difficulty has so far been addressed in three ways. Firstly, approximate methods can be applied that forego a consistency guarantee [10]. Secondly, tracks can be generated hierarchically from tracklets [3, 15, 6], akin to the use of superpixels in image processing. Thirdly, tracking is performed across pairs of frames only [11, 12, 19, 16].

Approach [3] is most similar to ours since it is–to our best knowledge–the only other global model of detections and assignments over many time steps. Therefore we chose this method to compare with.

3 Graphical Model for Global Tracking by Assignment

Both state space and assignment models treat detection (segmentation) and tracking as separate problems. We will focus on tracking in this article and assume given detections with a high misdetection rate ($\approx$ 10%).

Specifically, we propose a probabilistic model for the tracking problem that meets the requirements set out in the introduction and can give limited feedback to the detection step by incorporating the possibility of spurious detections, i.e. allows to switch off erroneous detection hypotheses to address the high misdetection rate. The model takes the form of a *chain graph* [9]—a directed graphical model (Bayesian network) of "supernodes", each of which consists of a conditional random field over a set of "subnodes". A chain graph model turns out to be necessary because neither Bayesian networks nor Markov random fields alone are compatible with the independence assumptions imposed by tracking.

Our model contains two types of random variables: *detection* variables and *assignment* variables. A binary detection variable $X_i^{(t)}$ is associated with the ith detected object (cell candidate) at time step t, and $\mathcal{X}^{(t)}$ denotes the set of all detection variables at time t. Values of these variables determine if the corresponding detections are accepted as a true object (cell) (and hence incorporated into the tracking interpretation), or rejected as a false alarm.[1] Binary assignment variables $Y_{ij}^{(t)}$ are associated with every pair of cell candidates i, j at times $t \in [1, \ldots, T-1]$, $t+1$, and $\mathcal{Y}^{(t)}$ is the set of all assignment variables between

[1] We deliberately do not model missed detections (i.e. objects, that are not visible in the data) since they are very rare in the 3d images of typical applications.

steps t and $t+1$. A value of 1 expresses the belief that cell candidate j at time step $t+1$ is identical with, or a child of, cell candidate i at time step t, and a value of 0 says that i and j are unrelated.

In addition, a number of natural consistency constraints are imposed on these variables by our application in developmental biology: First, each cell must have at most one predecessor, i.e. $\sum_i Y_{ij}^{(t)} \leq 1$. Second, each cell must have a unique fate—if it does not disappear (by leaving the data frame or other reasons), it can either move to (be assigned to) a unique cell candidate in the next time frame, or it can divide and be associated with exactly two cell candidates in the next time frame. These children, in turn, may not have any other nonzero incoming assignment variables. Third, there is a biological upper bound on the distance traveled between time frames, excluding all assignments that are too far apart. This leads to a significant reduction in the number of required assignment variables.

These consistency constraints define which configurations are impossible, i.e. have zero probability. In the next two subsections, we describe how the probabilities of feasible configurations are defined by connecting variables into a chain graph whose factors encode our knowledge about the plausibility of different trackings as a function of data-dependent properties such as distance traveled, probability of being a misdetection, etc.

Conditional random field over assignment variables Let us pretend for the moment that the optimal values of the detection variables are already known. Then the probability of a configuration $\mathcal{Y}^{(t)}$ of assignment variables connecting frames t and $t+1$ *given* the detection variables $\mathcal{X}^{(t)}$ and $\mathcal{X}^{(t+1)}$ can be expressed with an undirected graphical model, in particular a conditional random field (CRF):

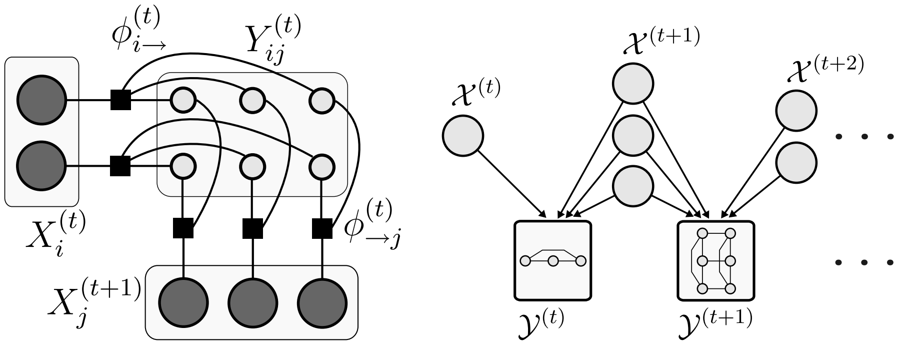
$$\mathrm{CRF}^{(t)}: \; P^{(t)}(\mathcal{Y}^{(t)} \mid \mathcal{X}^{(t)}, \mathcal{X}^{(t+1)}) \tag{1}$$

where

$$P^{(t)}(\mathcal{Y}^{(t)}|\mathcal{X}^{(t)}, \mathcal{X}^{(t+1)}) = \frac{1}{Z^{(t)}} \prod_{X_i^{(t)} \in \mathcal{X}^{(t)}} \phi_{i\rightarrow}^{(t)}(X_i^{(t)}, \mathcal{Y}_{i\rightarrow}^{(t)}) \prod_{X_j^{(t+1)} \in \mathcal{X}^{(t+1)}} \phi_{\rightarrow j}^{(t)}(\mathcal{Y}_{\rightarrow j}^{(t)}, X_j^{(t+1)}), \tag{2}$$

and $\phi_{i\rightarrow}^{(t)}$ and $\phi_{\rightarrow j}^{(t)}$ denote the factors for outgoing and incoming transitions respectively. Note that the CRF partition function $Z^{(t)}(\mathcal{X}^{(t)}, \mathcal{X}^{(t+1)})$ is only calculated over the assignment variables, since the detection variables are considered as given. A factor graph [14] representation of eq. (2) is shown in Fig. 3a. Each of the maximal cliques of the underlying undirected graph corresponds to a factor in the CRF. These CRFs will form the supernodes of the chain graph, while the individual assignment variables are the subnodes.

Analyzing the graph with d-separation the implicit independence assumptions of the CRF can be identified: an assignment $Y_{ij}^{(t)}$ is conditionally independent of all assignments $Y_{kl}^{(t)}$ involving different detection variables $k \neq i$, $l \neq j$,



(a) **Undirected part of the chain graph**. A single assignment CRF is shown as a factor graph [14]. The dark nodes represent fixed detection variables $\mathcal{X}$ (at t and $t+1$) and the smaller, light nodes assignment variables $\mathcal{Y}$.

(b) **Directed part of the chain graph**. The boxes symbolize supernodes containing assignment CRFs according to Fig. 3a.

Fig. 3: Chain graphical model for global tracking by assignment.

provided that all incoming assignments $\mathcal{Y}_{\to j}^{(t)}$ of detection j and all outgoing assignments $\mathcal{Y}_{i\to}^{(t)}$ of detection i are given:

$$Y_{ij}^{(t)} \perp\!\!\!\perp Y_{kl}^{(t)} \mid \mathcal{Y}_{\to j}^{(t)}, \mathcal{Y}_{i\to}^{(t)} \qquad l \neq j, i \neq k\,. \tag{3}$$

In other words, the assignment decisions only influence a small neighborhood of other assignment variables directly. By fixing enough variables the CRF can even decouple in two or more completely independent tracking problems—promoting the tractability of the chain graph model.

Combining assignment CRFs in a chain graph In order to get rid of the requirement that detection variables are already known, we connect all CRFs defined above by means of directed edges from detection variables to factors, in a manner analogous to a Bayesian network. The joint probability over all time steps can now be factorized as

$$P(\mathcal{X},\mathcal{Y}) = \prod_{t=1}^{T} \prod_{X_i^{(t)}\in\mathcal{X}^{(t)}} P_{\text{det}}(X_i^{(t)}) \cdot \prod_{t=1}^{T-1} P^{(t)}(\mathcal{Y}^{(t)}|\mathcal{X}^{(t)},\mathcal{X}^{(t+1)})\,, \tag{4}$$

where $P_{\text{det}}(X_i^{(t)})$ is the probability that a detection hypothesis should be accepted into the tracking. The corresponding chain graphical model is sketched in Fig. 3b. By treating each CRF as a single random variable (supernode) with as many states as configurations of the assignment variables (subnodes), d-separation can be easily applied to the directed part of the model. The (unobserved) supernodes have only incoming edges and are blocking the path between detection variables, revealing the following conditional independence relations:

1. Detection variables are independent from each other, i.e. $X_i \perp\!\!\!\perp (\mathcal{X} \backslash X_i) \ \ , \forall X_i$.
2. Consecutive assignment variables are conditionally independent given the connecting detection variables, i.e. $\mathcal{Y}^{(t)} \perp\!\!\!\perp \mathcal{Y}^{(t+1)} \mid \mathcal{X}^{(t+1)}$.

These are rather strong independence assumptions and–of course–only an approximation to the real world. Later, we introduce an extension (eqs. (11), (10)) that, on the one hand, improves the model accuracy but, on the other hand, weakens these relations and makes inference harder.

Inference by Integer Linear Programming Under the above model, the single most likely tracking is given by the maximum a-posteriori (MAP) configuration of variables. Finding this configuration can equivalently be restated as an energy minimization problem, by expressing the probability in eq. (4) as a Gibbs distribution $P(\mathcal{X}, \mathcal{Y}) = \frac{1}{Z} e^{-E(\mathcal{X}, \mathcal{Y})}$ such that

$$\underset{\mathcal{X},\mathcal{Y}}{\operatorname{argmax}}\, P(\mathcal{X}, \mathcal{Y}) = \underset{\mathcal{X},\mathcal{Y}}{\operatorname{argmin}}\, E(\mathcal{X}, \mathcal{Y})\,. \tag{5}$$

Rewriting the chain graph distribution in (4) as a sum of energies E allows us to apply integer linear programming (ILP) for minimization. This approach has two advantages. Firstly, it allows us to use state-of-the-art ILP solvers that are able to determine globally optimal solutions of (5). Any remaining shortcomings of our tracking results can thus be attributed to our model and are not a consequence of approximate optimization. Secondly, we can easily exclude biological impossible trackings by incorporating appropriate constraints in the linear program.

Application: Cell tracking Applying (5) to the chain graph model equations (2) and (4) gives the energy function

$$E(\mathcal{X}, \mathcal{Y}) = \sum_{t=1}^{T} \sum_{X_i^{(t)} \in \mathcal{X}^{(t)}} E_{\text{det}}(X_i^{(t)}) + \sum_{t=1}^{T-1} \left(\sum_i E_{\text{out}}(X_i^{(t)}, \mathcal{Y}_{i\rightarrow}^{(t)}) + \sum_j E_{\text{in}}(\mathcal{Y}_{\rightarrow j}^{(t)}, X_j^{(t+1)}) \right) \tag{6}$$

where E_{det}, E_{out}, and E_{in} are the energies corresponding to the factors P_{det}, $\phi_{i\rightarrow}^{(t)}$, and $\phi_{\rightarrow j}^{(t)}$ respectively.

The first term allows to incorporate evidence from the raw data regarding the presence or absence of a cell at a location where a potential target was detected. In our experiments, we use Random Forest [5] to estimate the probability $\hat{P}_{\text{det}}\left(X_i^{(t)}\right)$ of a detection being truly a cell, based on features such as intensity distribution and volume of each cell candidate found by the detection module. In keeping with the definition of the Gibbs distribution, we get

$$E_{\text{det}}\left(X_i^{(t)}\right) = \begin{cases} -\ln\left(\hat{P}_{\text{det}}\left(X_i^{(t)}\right)\right) & , X_i^{(t)} = 1 \\ -\ln\left(1 - \hat{P}_{\text{det}}\left(X_i^{(t)}\right)\right) & , X_i^{(t)} = 0 \end{cases} \tag{7}$$

The assignment energies E_{in} and E_{out} need to embody two kinds of prior knowledge: firstly, which trackings are consistent; and secondly, amongst all consistent trackings, which ones are plausible. To enforce consistency zero probability / infinite energy is assigned to cells dividing into more than two descendants, cells arising from the merging of two or more cells from the previous frame, and cell candidates that are regarded as misdetections.

To penalize implausible trackings, we need to parameterize our model appropriately. The model equations expose exactly five free parameters (C_{init}, C_{term}, C_{opp}, w, $\bar{d}$). C_{init} imposes a penalty for track initiation (i.e. when a cell (re-)appears in the data frame) and C_{term} for track termination (i.e. when a cell disappears from the data frame or is lost due to cell death). To discourage trivial solutions where too many objects are explained as misdetections, we exact a opportunity cost of C_{opp} for the lost opportunity to explain more of the data. Furthermore, each move is associated with a cost dependent on the squared length of that move. For cell divisions, finally, it is known that the descendants tend to be localized at an average distance $\bar{d}$ from the parent cell, and punish deviations from that expected distance.

All of the above can be summarized in the following assignment energies:

$$E_{\text{out}}(X_i^{(t)}, \mathcal{Y}_{i\rightarrow}^{(t)}) = \begin{cases} \infty & , X_i^{(t)} = 1 \wedge \sum_j Y_{ij}^{(t)} > 2 \quad \Big| \; > 2 \text{ children} \\ w\big((d-\bar{d})^2 + (d'-\bar{d})^2\big) & , X_i^{(t)} = 1 \wedge \sum_j Y_{ij}^{(t)} = 2 \quad \Big| \; \text{division} \\ wd^2 & , X_i^{(t)} = 1 \wedge \sum_j Y_{ij}^{(t)} = 1 \quad \Big| \; \text{move} \\ C_{\text{term}} & , X_i^{(t)} = 1 \wedge \sum_j Y_{ij}^{(t)} = 0 \quad \Big| \; \text{disappearance} \\ C_{\text{opp}} & , X_i^{(t)} = 0 \wedge \sum_j Y_{ij}^{(t)} = 0 \quad \Big| \; \text{opportunity} \\ \infty & , X_i^{(t)} = 0 \wedge \sum_j Y_{ij}^{(t)} > 0 \quad \Big| \; \text{tracked misdetection} \end{cases} \tag{8}$$

$$E_{\text{in}}(\mathcal{Y}_{\rightarrow j}^{(t)}, X_j^{(t+1)}) = \begin{cases} \infty & , X_j^{(t+1)} = 1 \wedge \sum_i Y_{ij}^{(t)} > 1 \quad \Big| \; > 1 \text{ parent} \\ 0 & , X_j^{(t+1)} = 1 \wedge \sum_i Y_{ij}^{(t)} = 1 \quad \Big| \; \text{move} \\ C_{\text{init}} & , X_j^{(t+1)} = 1 \wedge \sum_i Y_{ij}^{(t)} = 0 \quad \Big| \; \text{appearance} \\ 0 & , X_j^{(t+1)} = 0 \wedge \sum_i Y_{ij}^{(t)} = 0 \quad \Big| \; \text{misdetection, no parent} \\ \infty & , X_j^{(t+1)} = 0 \wedge \sum_i Y_{ij}^{(t)} > 0 \quad \Big| \; \text{misdetection with parent} \end{cases} \tag{9}$$

Note that more informative features could be extracted from the raw data: for instance, besides the length of a move, one could consider the similarity of the associated cell candidates to obtain an improved estimate of their compatibility. However, it is then no longer possible to select appropriate values for the parameters with a simple grid. Instead, a proper parameter learning strategy will be needed, which will be a subject of our future research.

Domain specific knowledge: minimal cell cycle length The model may be extended further by incorporating domain specific knowledge. For instance, cells must pass through specific cell cycle states (interphase, prophase, metaphase, anaphase, telophase) and thus, there is a biological lower bound for the duration of a cell cycle. In other words, it is impossible to observe a cell dividing twice within a number of subsequent frames as it is the case in Fig. 5a. This biological constraint can be integrated in our cell tracking model by expanding the detection variables $X_i^{(t)}$ to obtain the number of time steps since the last division of an individual cell, i.e. $X_i^{(t)} \in \{0, 1, ..., \tau\}$, where $X_i^{(t)} = 0$ still stands for misdetection and τ is a parameter for the minimal duration between division events of an individual cell. Hence, the energy functions given above change only slightly in that $X_i^{(t)} = 1$ becomes $X_i^{(t)} \geq 1$ and $X_i^{(t)} = \tau$ for the division energy. Besides, another two factors need to be introduced to incorporate the counting between successive detections. The corresponding energy functions are given by

$$E_{\text{cnt}\rightarrow}(X_i^{(t)}, \mathcal{Y}_{i\rightarrow}^{(t)}, \mathcal{X}_{i\rightarrow}^{(t+1)}) = \begin{cases} \infty \,, Y_{ij}^{(t)} = 1 \wedge \sum_k Y_{ik}^{(t)} = 2 \wedge X_j^{(t+1)} \neq 1 \\ \infty \,, Y_{ij}^{(t)} = 1 \wedge \sum_k Y_{ik}^{(t)} = 1 \wedge X_i^{(t)} < \tau \wedge X_j^{(t+1)} \neq X_i^{(t)} + 1 \\ \infty \,, Y_{ij}^{(t)} = 1 \wedge \sum_k Y_{ik}^{(t)} = 1 \wedge X_i^{(t)} = \tau \wedge X_j^{(t+1)} \neq \tau \\ 0 \quad , \text{otherwise} \end{cases} , \tag{10}$$

$$E_{\text{cnt}\leftarrow}(X_j^{(t+1)}, \mathcal{Y}_{\rightarrow j}^{(t)}) = \begin{cases} \infty \,, \sum_k Y_{kj}^{(t)} = 0 \wedge X_j^{(t+1)} \neq 0 \wedge X_j^{(t+1)} \neq \tau \\ 0 \quad , \text{otherwise} \end{cases} , \tag{11}$$

where $\mathcal{X}_{i\rightarrow}^{(t+1)}$ are the detection variables connected to $X_i^{(t)}$ through $\mathcal{Y}_{i\rightarrow}^{(t)}$. Technically, these rules are implemented as hard constraints on indicator variables each representing a possible state of the detection variables. It should be noted that with this modification, the detection variables are no longer independent. For that reason and due to the increased number of variables, computation time increases by a factor of 10.

4 Experiments

We applied the chain graph model to track cells during early embryogenesis of zebrafish (*Danio Rerio*) and fruit fly (*Drosophila*): the GFP-stained nuclei were imaged with different light sheet fluorescence microscopes.

4.1 Zebrafish Dataset

Keller et al. [13] recorded time-lapse volumetric images of developing zebrafish embryos (3d+t). A first cell lineage reconstruction of the blastula period based on that data was reported by [16]. Its segmentation-based cell detection method has high recall (false negative rate of 4±1% for cell detection) but suffers in precision due to segmentation errors and noise. Cell tracking was done by matching the time slices with minimal cost by integer linear programming. The test dataset consisted of 80 frames of the animal pole from 66 cells up to ca. 2400 cells. Its low segmentation precision warrants the application of the chain graph, which explicitly models noisy detections.

We implemented the method of Lou et al. [16] and replicated its benchmark for tracking the cells in the zebrafish dataset. The benchmark measures performance in terms of precision, recall, and f-measure[2] of the reconstructed cell

[2] precision: true pos./(true pos. + false pos.); recall: true pos./(true pos. + false neg.); f-measure: harmonic mean of precision and recall

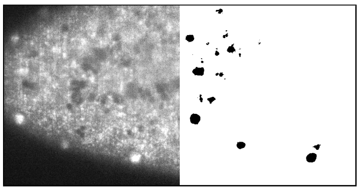


Fig. 4: **Drosophila dataset**: synopsis of raw data and corresponding segmentation; shown is a detail of slice 33 at time step 7.

Table 1: **Tracking results on the Zebrafish dataset**: our chain graph model shows improved performance in terms of f-measure compared to [16].

	Lou et al.[16]	Chain Graph
precision	0.807	**0.897**
recall	**0.861**	0.850
f-measure	0.833	**0.873**

movements and divisions between two time slices against a manually obtained tracking on the first 25 time slices. We use the segmentation result of [16] for a fair comparison of the two tracking methods.

4.2 Drosophila Dataset

The other dataset consists of 3d image stacks of a fruit fly syncytial blastoderm (cf. Fig. 1) over 40 time steps. We segmented the cells with the freely available ILASTIK software [22]. Subsequently we obtained a manual tracking to benchmark the automatic methods.

Compared to the zebrafish dataset the fruit fly dataset has lower contrast resulting in more ill-shaped cell segmentations and false positive detections that are hard to distinguish from the actual cells (see Fig. 4). On the other hand, cells reside only near the surface of the embryo, lowering the chance of wrong assignments to neighboring cells compared to the zebrafish embryo.

We examine the difference of our holistic approach over all time steps compared to a time slice by time slice tracking on that dataset. For the latter, we take the Random Forest predictions to filter out false alarms beforehand (0.5 probability threshold), add detection variables for all remaining objects to the graphical model and fix them as 'active'. Under these conditions the single assignment random fields are independent and the model reduces to a time step-wise tracking approach. Finally, we implemented the method of Bise et al. [3] and evaluated it with the same dataset and benchmark.

4.3 Implementation

We implemented the chain graph model in C++ using the OpenGM [1] library and wrote a MAP inference backend based on CPLEX.[3]

Parameter values are set according to an optimal f-measure of successfully reconstructed move, division, appearance, and disappearance events relative to a manually obtained tracking. We initialize the parameters with a reasonable setting and fine-tune them with a search on a grid in parameter space.

[3] A commercial (integer) linear programming solver. Free for academic use.

Table 2: **Tracking results on the Drosophila dataset**: a timestep-wise tracking with previously filtered misdetections (first col.) is inferior to our full chain graph model optimized over all time steps (second col.). The third column shows the results of the extended chain graph with the condition that division events of a cell must at least be 3 time steps apart from each other. Finally, the method of [3] (fourth col.) shows decent recall but suffers in precision.

	Fixed Detections	**Unconditioned Chain Graph**	**Chain Graph with $\tau = 3$**	**Bise et al.[3]**
precision	0.889	0.953	**0.956**	0.550
recall	0.933	0.957	**0.960**	0.718
f-measure	0.911	0.955	**0.958**	0.623

Runtimes on the presented datasets are of the order of minutes—obviating the need for approximate inference at the current problem size.

Finally, we implemented the methods by [16] and [3] in `C++` and found optimal parameters via grid search (improving over the ad hoc parametrization approach of [3]). Source code and parameters are provided as a supplementary.

4.4 Results

We perform experiments on two datasets. First, we compare our tracker against a previously published method [16] on the zebrafish dataset. The results are summarized in table 1. Second, we investigate the performance of the model on the Drosophila dataset, comparing a chain graph having fixed detection variables with an unconditioned chain graph and a chain graph with 4-state detection variables satisfying a minimal duration of 3 time steps between division events of a particular track. Additionally we put the chain graph model in perspective by comparing with a state-of-the-art cell tracking method by Bise et al. [3]. The tracking results are presented in table 2.

Finally, table 3 shows the Drosophila results in terms of correctly and incorrectly identified cells for both variants of the chain graph in contrast to human annotations and the Random Forest classifier alone (which was used to fix the variables).

5 Discussion

The MAP solution of the chain graph is obtained from all time steps at once considering properties of complete tracks. In particular a track needs a minimal number of active transitions to overcome the track initialization and termination costs together with the cost for misdetections. Cells that are only shortly present–due to low signal to noise or due to entering/leaving the data frame–will consequently be labeled as misdetections. On the other hand, cells that were badly segmented and were classified as misdetections by the Random Forest may

Table 3: **Cell identification performance on Drosophila dataset**: compared to an object classifier, the chain graph recovers less actual cells (true positives), but finds significantly more misdetections (true negatives). The unconditioned chain graph and the variant with a minimal cell cycle length don't differ significantly in identification performance (but in tracking performance: see table 2).

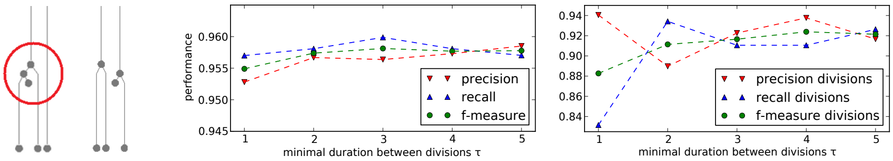
	Classifier	**Unconditioned Chain Graph**	**Chain Graph with $\tau = 3$**	**Human**
true positive	12391	12330	12346	12493
true negative	1284	1607	1598	1810
false negative	102	163	147	-
false positive	526	203	212	-

be set active as long as they reasonably continue a track (see Fig. 6 for an illustration of this behavior). The chain graph model can therefore be seen as a regularizer on the pure classifier output that trades some true positives (decreasing recall) against a significantly reduced number of false positives (increasing precision).

The above interpretation is supported by the results. Compared to the performance of method [16] on the zebrafish dataset, our method loses 1.1 percentage points (pp) in recall but gains 9pp in precision leading to an overall better performance. More insight into this precision-recall trade-off can be gained by looking at the cell vs. misdetection identification performance. Compared to the classifier alone, the (unconditioned) chain graph retrieves 61 less cells (less true positives), but labels 323 more misdetections correctly (more true negatives). (The chain graph controlling for minimal cell cycle length exposes the same behavior with nonsignificant differences in performance.)

Since correct cell identifications are a precondition for a successful recovery of move, division, appearance, and disappearance events, the higher object identification rate should directly transfer to the tracking performance. The results on the zebrafish are supporting this assumption, but could also be caused by the fact that our method employs a Random Forest classifier whereas the method of [16] just treats every segmented object as a cell. To exclude the influence of the random forest we examined another variant of the chain graph model, where we fix the detection variables as a preprocessing step using the very same Random Forest predictions. Compared to the chain graph with previously fixed detection variables, the full (unconditioned) model gains 2.4pp and 4pp in terms of recall and precision, respectively. This is clear evidence that the performance gain over the previously published step-by-step method is not only caused by the Random Forest classifier, but also by the holistic chain graph model that is optimized over many time steps at once.

Further improvements can be obtained by requiring a minimal time between two divisions. Table 2 shows an increase in f-measure by 0.3pp for a minimal temporal distance of three ($\tau = 3$) compared to the unconditioned model. At a first



(a) Impossible division

(b) Performance on Drosophila depending on τ

Fig. 5: **Extending the cell tracking model by the requirement of a minimal duration between division events**: (a) shows a biologically implausible tracking (left) and the result of our model when requiring a minimal duration of $\tau = 3$ (right). Performance results on the Drosophila dataset with varying τ are depicted in (b): shown is the performance for all events (left panel) and for division events only (right panel).

glance the gain seems nonsignificant, but the f-measure puts the same importance on every type of tracking event. Since the number of move events exceeds the number of divisions by a factor of 20, it is dominated by the former. Yet, the f-measure for divisions (exclusively) improves from 0.883 (precision 0.941, recall 0.832) to 0.917 (precision 0.923, recall 0.911). This is a significant gain and very important for an accurate assessment of cell ancestry since a single mistracked division can spoil a whole subsequent lineage. In practice, setting τ to a low value will already give a boost in performance with an acceptable overhead. The effect on the overall and the division performance when varying parameter τ is depicted in Fig. 5b. The method of Bise et al. [3] shows convincing results in regions with high data quality (cf. Fig. 2). However, its f-measure is 33pp worse compared to the chain graph. This is most likely caused by the high misdetection rate of 13% and can be seen from the low precision of 0.550 and Fig. 2.

Still the performance of our approach could be further improved. A manual inspection of incorrectly tracked divisions reveals that the tracker can be confused by divisions happening in close local proximity. A better model for divisions that also incorporates the change in shape and the geometric motion pattern—descendants move away from the ancestor in roughly opposite directions—could prevent such mistakes. Future datasets may contain cells that move in a more coordinated fashion. The presented move energies are optimal in case of Brownian motion, but will most likely underperform for directed motions. In that case the model can be extended with discrete momentum variables that are associated with the detected objects–analogous to our extension that controls the minimal cell cycle length. Finally, the model can easily be extended to consider merging objects (due to undersegmentation or 2d applications) by allowing more than one active incoming variable.

6 Conclusion

We presented a chain graph model for tracking by assignment over many time slices simultaneously. This model is highly robust against misdetections and

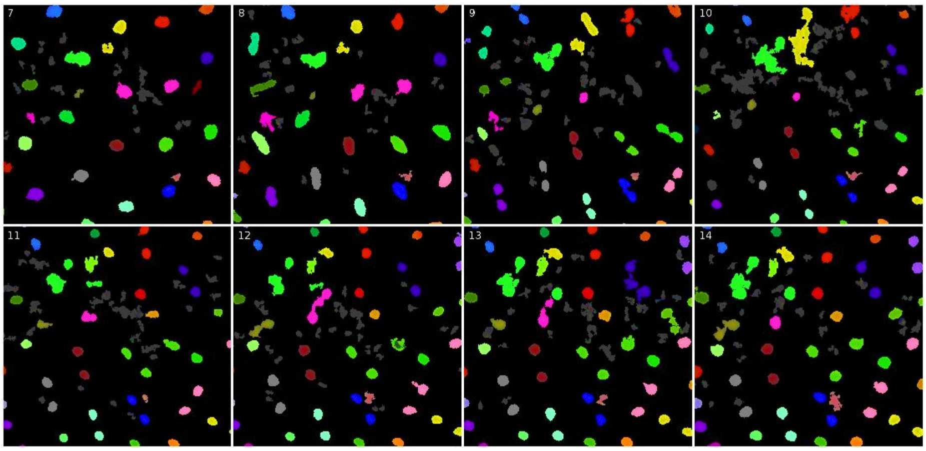


Fig. 6: **Sample tracking sequence obtained with the chain graph model**. The eight consecutive time steps show a detail of the Drosophila dataset projected to 2d. The colored spots are segmented objects, where objects with the same color are assigned to each other. Dark gray objects are misdetections as indicated by 'inactive' detection variables. Ill-shaped objects will be treated as active, if they are bridging two tracks in a sensible manner. On the contrary, cell-shaped objects may be labeled inactive, if they do not constitute a track of a certain length.

can consider temporal domain knowledge. It has been applied in the context of 3d+t cell tracking and achieved better performance than two recently published methods ([16], [3]). Furthermore, it showed that globally optimal tracking by assignment over many time slices is superior to the popular time step-wise approach. The performance could be further improved by introducing better motion models for cell movements and divisions.

Acknowledgments This work has been supported by the German Research Foundation (DFG) within the programme “Spatio-/Temporal Graphical Models and Applications in Image Analysis”, grant GRK 1653.

References

1. Andres, B., Kappes, J.H., Koethe, U., Schnoerr, C., Hamprecht, F.A.: An empirical comparison of inference algorithms for graphical models with higher order factors using opengm. In: Goesele, M., Roth, S., Kuijper, A., Schiele, B., Schindler, K. (eds.) Pattern Recognition, LNCS, vol. 6376, pp. 353–362. Springer (2010)
2. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. Signal Processing, IEEE Transactions on 50(2), 174 –188 (2002)
3. Bise, R., Yin, Z., Kanade, T.: Reliable cell tracking by global data association. In: IEEE International Symposium on Biomedical Imaging (ISBI 2011) (2011)
4. Blake, A., Kohli, P., Rother, C. (eds.): Markov Random Fields for Vision and Image Processing. MIT Press (2011)

5. Breiman, L.: Random forests. Machine Learning 45, 5–32 (2001)
6. Brendel, W., Amer, M., Todorovic, S.: Multiobject tracking as maximum weight independent set. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011). pp. 1273 –1280 (2011)
7. Doucet, A., Johansen, A.: A tutorial on particle filtering and smoothing: Fifteen years later. In: Handbook of Nonlinear Filtering. Oxford University Press (2011)
8. Fox, E., Choi, D., Willsky, A.: Nonparametric Bayesian methods for large scale multi-target tracking. In: Signals, Systems and Computers, 2006. ACSSC'06. Fortieth Asilomar Conference on. pp. 2009–2013 (2006)
9. Frydenberg, M.: The chain graph Markov property. Scandinavian Journal of Statistics 17, 333–353 (1990)
10. Jiang, H., Fels, S., Little, J.J.: A linear programming approach for multiple object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007) (2007)
11. Kachouie, N.N., Fieguth, P.W.: Extended-Hungarian-JPDA: exact single-frame stem cell tracking. IEEE Transactions on Bio-Medical Engineering 54 (2007)
12. Kanade, T., Yin, Z., Bise, R., Huh, S., Eom, S., Sandbothe, M.F., Chen, M.: Cell image analysis: Algorithms, system and applications. In: IEEE Workshop on Applications of Computer Vision (WACV) (2011)
13. Keller, P.J., Schmidt, A.D., Wittbrodt, J., Stelzer, E.H.: Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. Science 322, 1065–1069 (2008)
14. Kschischang, F., Frey, B., Loeliger, H.A.: Factor graphs and the sum-product algorithm. Information Theory, IEEE Transactions on 47, 498 –519 (2001)
15. Li, Y., Huang, C., Nevatia, R.: Learning to associate: HybridBoosted multi-target tracker for crowded scene. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009) (2009)
16. Lou, X., Kaster, F.O., Lindner, M.S., Kausler, B.X., Koethe, U., Jaenicke, H., Hoeckendorf, B., Wittbrodt, J., Hamprecht, F.A.: DELTR: Digital embryo lineage tree reconstructor. In: IEEE International Symposium on Biomedical Imaging (ISBI 2011) (2011)
17. Meijering, E., Dzyubachyk, O., Smal, I., van Cappellen, W.A.: Tracking in cell and developmental biology. Seminars in Cell & Developmental Biology 20 (2009)
18. Melani, C., Peyrieras, N., Mikula, K., Zanella, C., Campana, M., Rizzi, B., Veronesi, F., Sarti, A., Lombardot, B., Bourgine, P.: Cells tracking in a live zebrafish embryo. In: IEEE Engineering in Medicine and Biology Society (EMBS 2007) (2007)
19. Padfield, D., Rittscher, J., Roysam, B.: Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. Medical Image Analysis 15(4) (2011)
20. Padfield, D., Rittscher, J., Thomas, N., Roysam, B.: Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. Medical Image Analysis 13(1), 143–155 (2009)
21. Smal, I., Meijering, E., Draegestein, K., Galjart, N., Grigoriev, I., Akhmanova, A., Van Royen, M., Houtsmuller, A., Niessen, W.: Multiple object tracking in molecular bioimaging by Rao-Blackwellized marginal particle filtering. Medical Image Analysis 12(6), 764–777 (2008)
22. Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.A.: ILASTIK: Interactive learning and segmentation toolkit. In: IEEE International Symposium on Biomedical Imaging (ISBI 2011) (2011)
23. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys (CSUR) 38 (2006)