

Conservation Tracking

Martin Schiegg¹

Philipp Hanslovsky¹

Bernhard X. Kausler¹

Lars Hufnagel²

Fred A. Hamprecht¹

¹ University of Heidelberg, IWR/HCI
69115 Heidelberg, Germany

firstname.lastname@iwr.uni-heidelberg.de

² European Molecular Biology Laboratory (EMBL)
69117 Heidelberg, Germany

hufnagel@embl.de

Abstract

The quality of any tracking-by-assignment hinges on the accuracy of the foregoing target detection / segmentation step. In many kinds of images, errors in this first stage are unavoidable. These errors then propagate to, and corrupt, the tracking result.

Our main contribution is the first probabilistic graphical model that can explicitly account for over- and undersegmentation errors even when the number of tracking targets is unknown and when they may divide, as in cell cultures. The tracking model we present implements global consistency constraints for the number of targets comprised by each detection and is solved to global optimality on reasonably large 2D+t and 3D+t datasets. In addition, we empirically demonstrate the effectiveness of a postprocessing that allows to establish target identity even across occlusion / undersegmentation. The usefulness and efficiency of this new tracking method is demonstrated on three different and challenging 2D+t and 3D+t datasets from developmental biology.

1. Introduction

The tracking of multiple *dividing* targets is a challenging computer vision problem and has useful application e.g. in life science [10] or in the car industry [13]. Due to the occurrence of object divisions at any time, the number of targets for each time step is unknown even if user-specified for a subset of frames.

Multi-object tracking in general may be implemented as a two-step pipeline consisting of a detection/segmentation step and a data association or assignment/tracking step [18]. Such approaches, however, are obviously susceptible to errors in the detection step which are propagated to the tracking model and typically cannot be corrected downstream. Therefore, the ultimate goal of data association tracking is

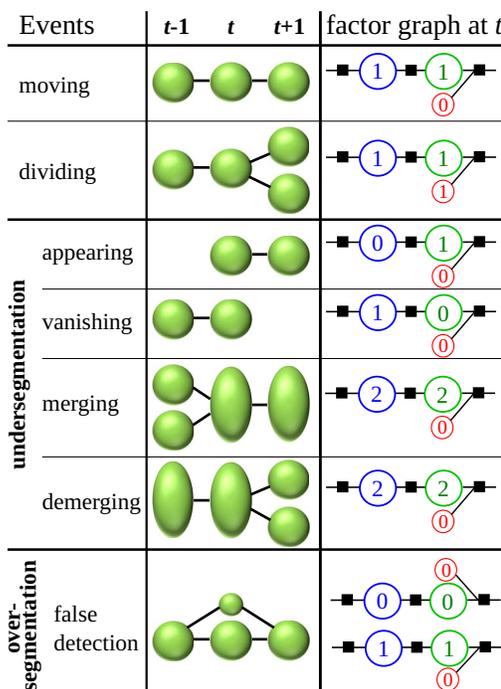


Figure 1: The proposed tracking-by-assignment model accounts for all of these events. Left column: Objects (represented as balls) are associated (edges) with each other over three time steps. Right: Excerpt of the proposed factor graph showing the three detection variables for the connected component at time t : Red variables are indicators for a division event. The other variables, taken together, represent the number of targets covered by a detection but they can also represent the other depicted scenarios such as disappearance or “demerging”. See Fig. 3 for more details.

to address detection and data association *jointly* such that both steps can maximally benefit from each other and information can be propagated from more to less obvious parts of

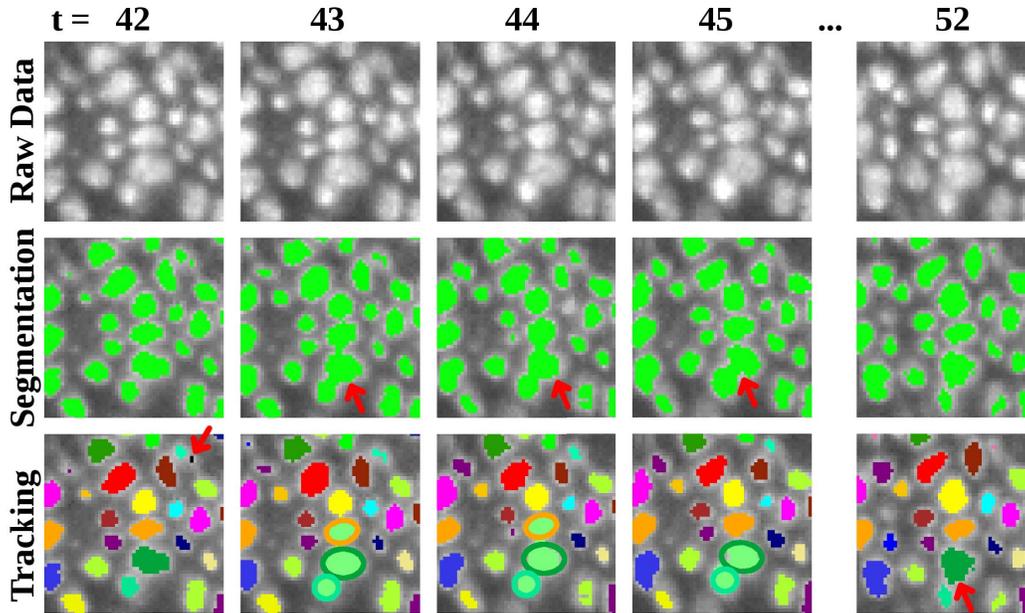


Figure 2: Tiny excerpt of dataset B with its almost indistinguishable objects. A short sequence of the raw data is depicted as 2D slices (top row) from 3D+time data and displays cells in a developing *Drosophila* embryo. Due to low contrast, multiple cells are segmented as only one connected component (undersegmentation) as pointed out in the middle row. Our tracking model (bottom row) can handle such errors and preserves the target identities as indicated by colors (see the three previously merged cells in $t = 52$) by fitting the correct number of Gaussians (ellipses) to detections containing multiple objects. Furthermore, the proposed factor graph can handle false detections (oversegmentation) as indicated by the black detection in frame 42 (bottom row).

the data. There are first approaches addressing joint detection and tracking [16, 17], but none of them has been extended to deal with *dividing* objects. Given that the tracking of multiple dividing objects already is an NP-hard problem [14] in itself, joint detection and assignment is harder still.

As a first step into this direction, we propose a model that handles detection errors *explicitly* in the tracking step and can even correct most of them. Typical segmentation errors are depicted in the lower rows in Fig. 1 and can be categorized into over- and undersegmentation errors occurring due to low contrast or noise in the images. Furthermore, a real data example is given in Fig. 2. Oversegmentation may result in false detections whereas undersegmentation could lead to the appearance and vanishing of tracks or to accidental track merging. In this context, the divisibility of the objects is particularly challenging since demerging due to previous merging must be distinguished from object division. Note that we will differentiate between object *division* and object *demerging* throughout the paper.

We present the first method which explicitly models *all* of the potential *segmentation errors* outlined above in one probabilistic graphical model. The proposed factor graph models conservation laws for the number of objects con-

tained in each detection to ensure *global consistency* of the solution. In other words, the model not only assures consistency between pairs of frames but can also resolve segmentation errors which only become evident from considering a complete time series at once (cf. Fig. 2). In this way, temporarily merged targets can be *resolved under identity preservation* even for objects which are merged during longer sequences. For this purpose, a spatial Gaussian mixture model with the appropriate number of components is fitted to the undersegmented regions. Object properties such as velocity are easy to represent in state space models, but notoriously difficult to model in a tracking-by-assignment approaches. To avoid bias towards slow motions and false assignments during group movement, we *estimate group motion* in a preprocessing step, using patchwise cross correlation.

The remainder of this paper is structured as follows. We commence with the review of prior art and propose the tracking framework – and particularly the construction of the factor graph – in Sec. 3. Before we conclude in Sec. 5, we present and discuss experiments on 2D+t and 3D+t data in Sec. 4 which showcase the effectiveness of the proposed approach.

2. Related Work

Existing tracking approaches can broadly be categorized into three: (i) space-time segmentation, (ii) state space models, and (iii) tracking-by-assignment. The first is only applicable at frame rates that make for small, ideally sub-pixel, displacements of objects between subsequent images.

State space models or Bayesian sequential filtering are not easily applicable to an unknown or variable number of objects, calling for costly strategies such as reversible jump MCMC or Gibbs sampling on Dirichlet process mixture models [4]. In such a setup, dealing with *divisible* objects is harder still.

Tracking-by-assignment gracefully handles multiple, and even dividing objects; on the downside, object properties such as object velocity need to be implemented using factors that are higher order in time.

We draw inspiration from, and build on, a series of excellent papers. The tracking of undersegmented objects was first described in [11] and soon extended to deal with fragmentation (false positive detections) [2]. Shitrit *et al.* [1] additionally address object identity preserving for possible occlusions of objects by exploiting global appearance constraints. Furthermore, the authors in [8] account for both dividing objects and undersegmentation, and exploit local evidence in pairs of frames to find undersegmented objects. Their model, however, does not guarantee consistency over all time steps and detections.

The structure of our graphical model also builds on the network flow formulation in [19]. Note, however, that allowing for object division no longer permits to do inference via an ordinary network flow computation as in [19]. Instead, admitting divisions necessarily turns the problem into an integer flow problem with homologous arcs (i.e. flow along separate edges is required to be identical) and a constraint matrix which is not totally unimodular. Hence, there is no guarantee to obtain integer solutions in a network flow, and rounding [12] gives only approximate solutions to a problem that is proven to be NP-hard [14].

Moreover, the only model which handles the tracking of dividing objects in a global probabilistic framework is the graphical model presented in [5]. While oversegmentation is addressed in terms of false detections, it cannot deal with undersegmentation such as merged objects.

3. Tracking Divisible Objects in spite of Over- and Undersegmentation

The purpose of this work is to track dividing objects based on an error-prone segmentation. We therefore model data association in a probabilistic graphical model [6] where we explicitly handle over- and undersegmentation errors (cf. Fig. 1). Here, the key idea is that all detections over all time steps are handled simultaneously in a holistic graphi-

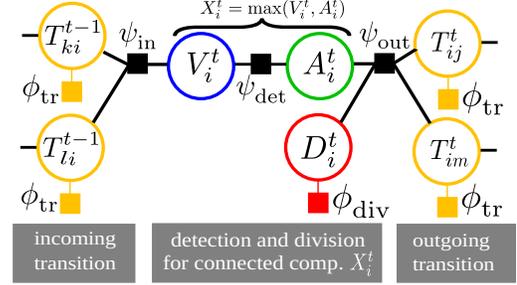


Figure 3: Factor graph for one detection X_i^t with two incoming and two outgoing transition candidates: One detection X_i^t is represented by two multi-state variables, V_i^t and A_i^t , to allow for vanishing and appearance, respectively. Furthermore, the binary variable D_i^t indicates whether object X_i^t is about to divide. See Fig. 1 for different configurations of these variables. Moreover, transition variables $T \in \{0, \dots, m\}$ indicate how many objects are associated between two respective detections. Here, the black squares implement conservation laws, i.e. the sum of the left-hand side must equal the sum of the right-hand side, whereas colored squares represent unary factors of the variables.

cal model on which inference is performed globally. In this way, each segmented region is assigned the number of objects it contains while conservation laws across subsequent detections guarantee *global* consistency. Finally, each detection is partitioned into its inferred number of objects by fitting a Gaussian mixture model such that post-hoc linking yields identity preservation for temporarily merged targets. It should be noted that we distinguish between the terms *object* and *detection* which denote one target and one connected component, respectively, where a detection may comprise multiple objects. In the following, we describe our tracking workflow in detail for which a schematic overview is depicted in Fig. 4.

3.1. Graphical Model Implementing Global Conservation Laws

We design a factor graph [7], illustrated in Fig. 3, which contains three categories of variables: *Detection* variables (comprising appearance and vanishing variables) for each connected component X_i^t from the segmented image, *division* variables which indicate whether an object is about to divide, and *transition* variables that associate detections in two adjacent time frames with each other.

In particular, each detection X_i^t is represented by an appearance variable $A_i^t \in \{0, \dots, m\}$ and a vanishing variable $V_i^t \in \{0, \dots, m\}$, where m is the maximal number of objects contained in one detection. The number of objects comprised by detection X_i^t is given by their maximum

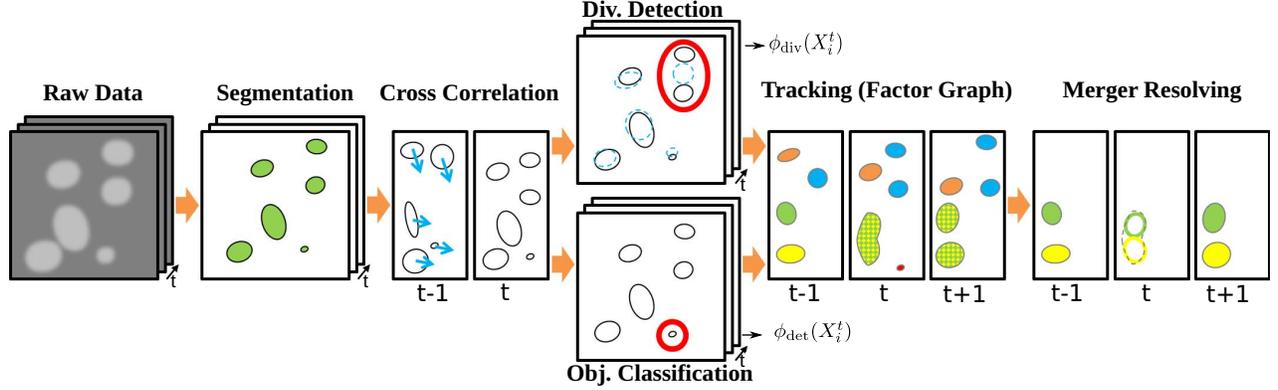


Figure 4: Objects are first detected from raw data by segmentation. Subsequently, on pairs of frames, patch-wise cross correlation on the binary images yields rough estimates about the displacement of groups of objects. Following this, probabilistic classifiers determine the unary potentials of each detection, i.e. they estimate the division probability and a probability mass function of the number of objects contained in each detection. These potentials are then used in the proposed factor graph (cf. Fig. 3) to find a *globally consistent* tracking solution (here, tracks are indicated by colors). In the last step, detections which were found to contain more than one object (yellow/green in this example) are partitioned by fitting a spatial Gaussian mixture model with the respective number of kernels and the demerged objects are being tracked again in order to preserve their original identities.

$X_i^t = \max(V_i^t, A_i^t)$.¹ The appearance and vanishing variables of one detection are connected by $\psi_{\text{det}}(A_i^t, V_i^t, f_i^t) =$

$$= \begin{cases} -\ln \left(\hat{P}(X_i^t = k | f_i^t) \right), & V_i^t = A_i^t = k \\ -\ln \left(\hat{P}(X_i^t = k | f_i^t) \right) + kw_{\text{app}}, & V_i^t = 0, A_i^t = k > 0 \\ -\ln \left(\hat{P}(X_i^t = k | f_i^t) \right) + kw_{\text{van}}, & V_i^t = k > 0, A_i^t = 0 \\ \infty, & \text{otherwise} \end{cases} \quad (1)$$

where $k \in \{0, \dots, m\}$.

In other words, only three different kinds of configurations are allowed (cf. Fig. 1): $V_i^t = A_i^t = k$ indicates that X_i^t comprises k objects (and X_i^t is a false detection if $k = 0$); $V_i^t = 0, A_i^t > 0$ means that the object(s) in X_i^t is/are appearing in this time step (i.e. starting a new track); whereas $V_i^t > 0, A_i^t = 0$ stands for their disappearance at time t .

Here, the design parameters w_{app} and w_{van} penalize spontaneous appearance and vanishing. $P(X_i^t = k | f_i^t)$ is determined by a probabilistic discriminative classifier where f_i^t stands for local evidence. In our experiments, we train a random forest [3] on local features such as the size of the connected component, its mean and variance of intensity, or the standard deviation along the principal components of the detected segment.

Now, we introduce a binary variable D_i^t which indicates whether an object in detection X_i^t is about to divide or not. Again, its unary potential is determined by a probabilistic

¹Note that X_i^t is given by a deterministic function of A_i^t, V_i^t and is hence omitted from the model in Fig. 3.

classifier based on local evidence. In our experiments, we deal with cell tracking and therefore utilize domain specific features for cell division. These include the angle the two nearest neighbors X_j^{t+1}, X_l^{t+1} at $t + 1$ enclose with X_i^t , the mean and variance intensity of X_i^t , and the ratios of the squared distances to X_j^t , the mean intensities, and the sizes of X_j^{t+1} and X_l^{t+1} . For all features where appropriate the region centers corrected by their cross correlation offset (cf. Sec. 3.3) are appended in addition. Division nodes are only added if the respective detection has at least two potential successors in the next time frame and the score from the division detection classifier is above some small threshold.

The third category of random variables in the proposed graphical model, the transition variables $T_{ij}^t \in \{0, \dots, m\}$, are added for pairs of detections X_i^t, X_j^{t+1} in two subsequent frames. Their value indicates the number of objects of X_i^t which are assigned to X_j^{t+1} (this can be some or all). Local evidence for pairs of detections X_i^t, X_j^{t+1} is injected by

$$\hat{P}(T_{ij}^t = k | d_{ij}^t) = \begin{cases} 1 - \exp\left(-\frac{d_{ij}^t}{\alpha}\right), & k = 0 \\ \exp\left(-\frac{d_{ij}^t}{\alpha}\right), & k \neq 0 \end{cases}, \quad (2)$$

where $d_{ij}^t = (\mathbf{x}_j^{t+1} - (\mathbf{x}_i^t + \mathbf{u}_i^t))^2$ is the squared distance of the region centers \mathbf{x} corrected by the estimated cross correlation offset \mathbf{u} (cf. Sec. 3.3), and α is a design parameter. It should be noted that by taking into account the *estimated* rather than the *detected* region centers, acceleration of an object is penalized instead of its velocity.

So far, only probabilistic *local* evidence has been implemented into the model. To ensure *global* consistency of the inferred solution, we augment the factor graph with conservation laws: In particular, the number of objects in detection X_i^t in t must equal to the sum of objects associated with X_i^t in time $t-1$ and $t+1$ (while taking object divisions into consideration). The conservation laws which guarantee these equivalences are implemented in the black squares in Fig. 3, where, broadly speaking, the sum of objects on the left-hand side must equal the sum of the right-hand side. For instance, the conservation law for the outgoing transitions of X_i^t is $\psi_{\text{out}}(A_i^t, T_{ij_0}^t, \dots, T_{ij_{n'}}^t) =$

$$= \begin{cases} \infty, & \sum_{l \in \{j_0, \dots, j_{n'}\}} T_{il}^t \neq A_i^t + D_i^t \\ \infty, & \exists l \in \{j_0, \dots, j_{n'}\} : T_{il}^t > A_i^t \\ \infty, & \sum_{l \in \{j_0, \dots, j_{n'}\}} T_{il}^t \neq 2 \text{ if } D_i^t = 1. \\ \infty, & A_i^t \neq 1 \text{ if } D_i^t = 1 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Furthermore, since sparse objects may lead to isolated (sub-)paths in the graphical model, i.e. paths where only one transition between two detections is possible, we subsume variables in such paths in *tracklets* and set their unary potential to the sum of the single detections' unaries plus their transition potentials for each possible configuration. In this way, whole tracklets may be treated just like single detections, which leads to major speed-ups in optimization.

Finally, the approximate maximum a-posteriori (MAP) solution of the proposed factor graph can be found using standard message passing algorithms. Alternatively, by minimizing the energy

$$\underset{A, V, D, T}{\operatorname{argmin}} E(A, V, D, T) = \underset{A, V, D, T}{\operatorname{argmin}} \sum_t \sum_i \left(E_{\text{det}}(A_i^t, V_i^t) + E_{\text{div}}(D_i^t) + \sum_j E_{\text{tr}}(T_{ij}^t) \right), \quad (4)$$

subject to an integrality constraint and the linear hard constraints implicitly given in the potentials ψ_{det} , ψ_{out} , and ψ_{in} , a solution can be obtained using integer linear programming (ILP) solvers. We opt for the latter since the problem can be solved exactly to global optimality for reasonably sized problems.

The energy terms in Eq. (4) are obtained by reformulating probabilities in the energy domain utilizing the well-known Gibbs distribution $P(X) = \frac{1}{Z} e^{-E(X)}$, where Z is a normalizing factor.

3.2. Resolving Merged Objects

The inferred result of the described factor graph yields the number of objects covered by one detection X_i^t and the

number of objects T_{ij}^t transferred between two detections X_i^t, X_j^{t+1} in adjacent time steps. Identities of individual objects are amalgamated into a cluster whenever undersegmentation leads to seeming mergers. To recover individual identities, we introduce the following model based on the inferred configuration of the factor graph.

Given the number of objects k contained in detection X_i^t , we fit a Gaussian mixture model with k normal distributions $\mathcal{N}(\mu_l, \Sigma_l)$ of unknown weight π_l to the connected component with pixels/voxels $\{x_1, \dots, x_n\}$, i.e. we maximize

$$P(x_1, \dots, x_n) = \prod_{j=1}^N \sum_{l=1}^k \pi_l P(x_j | \mu_l, \Sigma_l). \quad (5)$$

The resulting clusters with means $\mu_l, l \in \{1, \dots, k\}$ are then treated as separate detections with centers μ_l . Next, another factor graph as described in Sec. 3.1 is constructed comprising all newly resolved and all attaching original detections. We modify this *merger resolving factor graph* by setting all $A_i^t = V_i^t = 1$, i.e. in this post-processing step, we disallow appearance, vanishing, and false detections. This graphical model is again solved to global optimality and its solution hence preserves identities of objects, even for long sequences of merged objects.

3.3. Cross Correlation for Region Center Correction

Most tracking-by-assignment approaches penalize displacements of objects in terms of squared distance between objects of adjacent time frames. However, if a group of objects is moving rapidly in the same direction, this approach may lead to false assignments due to temporal aliasing. On this account, we adapt our model to penalize acceleration rather than velocity.

Before constructing the factor graph described in Sec. 3.1, we perform patch-wise cross correlation on the binary images on pairs of adjacent frames. In other words, for each patch $g^t(\mathbf{x})$ at time t of a user-specified size, we search for the best match in its neighborhood $h^{t+1}(\mathbf{x})$ at $t+1$ by maximizing

$$c(\mathbf{x}) = \sum_{\mathbf{x}} g^t(\mathbf{x}) h^{t+1}(\mathbf{x} - \mathbf{u}^t) \quad (6)$$

to estimate an offset \mathbf{u}^t for each pixel/voxel at t .

The transition prior ϕ_{tr} can then be computed based on the detection centers corrected by those offsets to find the displacement *relative* to the motion of the object's neighborhood.

3.4. Implementation

We have implemented the proposed algorithm in C++ and inference is performed using CPLEX. Optimization

	Overall: 12,289			Divisions: 380		
	Prec.	Rec.	F-meas.	Prec.	Rec.	F-meas.
Kausler <i>et al.</i> [5]	0.96	0.96	0.96	0.92	0.91	0.92
Classifiers only*	N/A	N/A	N/A	0.79	0.69	0.74
Ours ($m = 1$)	0.94	0.95	0.94	0.92	0.88	0.90

Table 1: Cell tracking results on dataset A: precision ($= \frac{TP}{TP+FP}$), recall ($= \frac{TP}{TP+FN}$), and f-measure ($= 2 \cdot \frac{\text{prec.} \cdot \text{rec.}}{\text{prec.} + \text{rec.}}$) for the overall pairwise events (move, appearance, disappearance, divisions) and divisions in particular. For a description of *Classifiers only*, refer to Tab. 2.

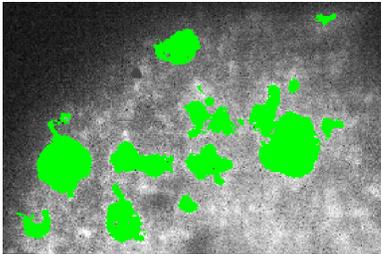


Figure 5: An excerpt of one time step of dataset A. Green color indicates detections including many false positives.

time is between one and 30 minutes on a current workstation, even for the experiments with large problem sizes presented in the following section. The source code together with a GUI for the complete workflow will be freely available on the authors’ website.

4. Experiments and Results

Cell tracking is a natural application for the tracking of dividing objects, particularly challenging due to their almost texture-less appearances, which makes them nearly indistinguishable from each other. Furthermore, microscope images often suffer from low contrast which typically makes segmentation error-prone. Especially in dense cell populations, undersegmentation is a common cause for errors.

In order to show the efficiency and accuracy of the proposed algorithm, we perform experiments on three challenging datasets, datasets A and B are 3D+t, dataset C is 2D+t. Note that, although both datasets A and B show developing embryos of fruit flies, they are of drastic difference in appearance in terms of contrast. Furthermore, the density of cell populations due to their diverging stages in the developmental process of the embryo are highly diverging.

In all experiments, we use random forests [3] each comprising 100 trees grown to purity as classifiers for cell number ϕ_{det} and cell mitosis, ϕ_{div} . Small training sets (≤ 30 samples for positive classes) are taken from the data. For a

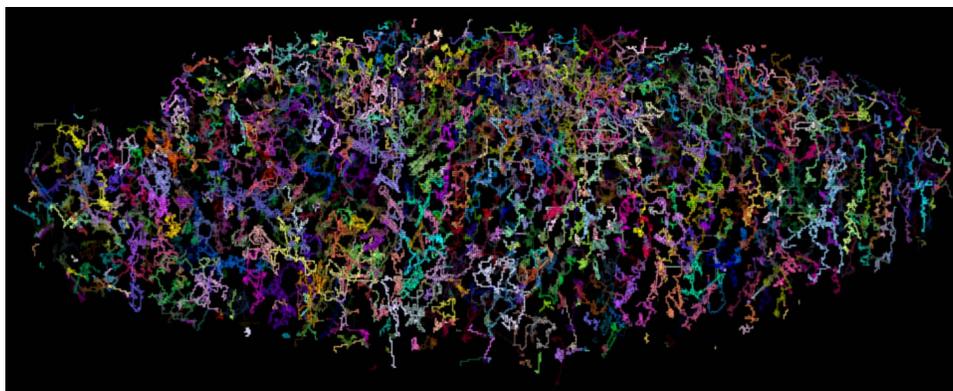
fair comparison, we used the same cell number classifier in our method and the competitive model.

First, we evaluate our model on the publicly available dataset from [5] which shows a *Drosophila* embryo in syncytial blastoderm (dataset A). Its segmentation (cf. Fig. 5) consists of about 300 connected components on average over 40 time steps of $2,362 \times 994 \times 47$ volumes and shows many false detections. We take the published segmentation of this dataset and its gold standard to compare with the recently published cell tracking model by Kausler *et al.* [5]. Their segmentation contains no merged objects and thus, we set in our model the maximal number of objects per detection to one, i.e. $m = 1$. In this experiment, we use the cell detection classifier of [5] and set our parameters to $\alpha = 25$, $w_{\text{app}} = 50$, $w_{\text{van}} = 50$, $w_{\text{tr}} = 13$, $w_{\text{div}} = 28$, where the latter two parameters weight the transition and division priors versus the detection prior. The results of this experiment are given in Tab. 1. Our model yields an f-measure of 0.94 taking all pairwise events, i.e. moves, appearances, disappearances, and divisions, together, comparable to [5] (0.96). The f-measure for divisions in [5] is slightly better than ours, namely 0.92 compared to 0.90, which is due to their model making assumptions about minimal durations between division events (cf. [5, Fig. 5]).

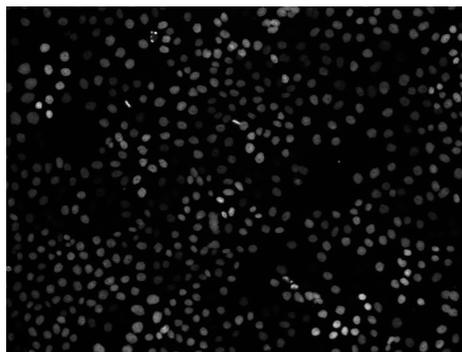
The second dataset (dataset B) again shows a *Drosophila* fruit fly, but this time shortly after gastrulation. Due to the embryonic development, the cell population is now much denser than in dataset A, resulting in a high number of undersegmented objects (cf. Fig. 2). Furthermore, different from dataset A, cells enter mitosis highly heterogeneously in time. In this dataset, on average ≈ 800 cells are tracked over 100 time steps of $730 \times 320 \times 30$ volumes. A gold standard for this dataset has been manually acquired. This dataset is segmented using the segmentation toolkit *ilastik* [15], a pixel-wise random forest classifier for segmentation. The design parameters in our factor graph – for the case of allowing maximally 4 cells in one detection (i.e. $m = 4$) – are set to $\alpha = 5$, $w_{\text{app}} = 100$, $w_{\text{van}} = 100$, $w_{\text{tr}} = 24$, $w_{\text{div}} = 36$. A 3D rendering of the resulting trajectories over all time steps is depicted in Fig. 6a. The results (cf. Tab. 2) show that our method outperforms the cell tracking model in [5]. As indicated by the division f-measure of 0.71 compared to 0.06 of the competitive model, the explicit modeling and distinction of demerging and dividing – together with the probabilistic division prior ϕ_{div} – brings a boost in the detection of mitotic events. Besides, due to the consideration of all detections of all frames in one holistic model and due to the conservation laws posed, our factor graph can accurately (precision of 0.78) detect the true number of targets contained in each detection. For this evaluation measure, only detections have been considered which contain more than one cell. Finally, with an f-measure of 0.68, our framework can resolve the original

		Moves			Divisions			Mergers			Resolved Mergers		
		Prec.	Rec.	F-measure	Prec.	Rec.	F-measure	Prec.	Rec.	F-measure	Prec.	Rec.	F-measure
Dataset B	Kausler <i>et al.</i> [5]	0.92	0.92	0.92	0.05	0.12	0.06	N/A	N/A	N/A	N/A	N/A	N/A
	Classifiers only*	N/A	N/A	N/A	0.83	0.64	0.72	0.63	0.31	0.41	N/A	N/A	N/A
	Ours ($m = 1$)	0.97	0.95	0.96	0.62	0.63	0.63	N/A	N/A	N/A	N/A	N/A	N/A
	Ours ($m = 2$)	0.97	0.97	0.97	0.53	0.79	0.64	0.71	0.54	0.61	0.72	0.61	0.66
	Ours ($m = 3$)	0.97	0.97	0.97	0.70	0.76	0.73	0.73	0.58	0.64	0.73	0.63	0.67
Ours ($m = 4$)	0.97	0.97	0.97	0.65	0.77	0.71	0.78	0.59	0.67	0.74	0.63	0.68	
Dataset C	Kausler <i>et al.</i> [5]	0.99	0.97	0.98	0.65	0.68	0.66	N/A	N/A	N/A	N/A	N/A	N/A
	Classifiers only*	N/A	N/A	N/A	0.92	0.56	0.70	0.41	0.62	0.49	N/A	N/A	N/A
	Ours ($m = 1$)	0.99	0.97	0.98	0.68	0.71	0.70	N/A	N/A	N/A	N/A	N/A	N/A
	Ours ($m = 2$)	1.00	0.99	0.99	0.85	0.76	0.80	0.73	0.60	0.66	0.79	0.70	0.74
	Ours ($m = 3$)	1.00	0.99	0.99	0.85	0.77	0.80	0.84	0.69	0.76	0.85	0.75	0.79
Ours ($m = 4$)	1.00	0.99	0.99	0.85	0.76	0.80	0.84	0.69	0.76	0.85	0.75	0.80	

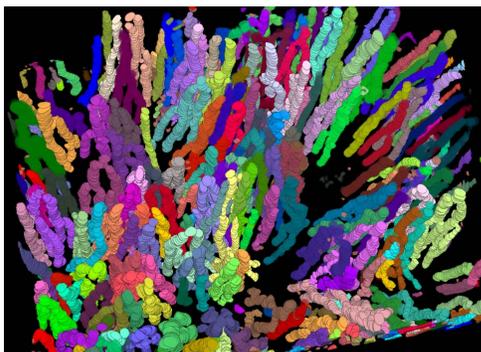
Table 2: Cell tracking results on datasets B and C: Our model with a different number of objects in one detection allowed ($m = 1$ to $m = 4$) can best handle the under-/oversegmentation errors occurring in these datasets. Here, merged objects are only counted as true positives if the true number (≥ 2) of objects in the connected component is found. Finally, *resolved mergers* indicates, how many of the merged objects have been resolved to their original identities after demerging. (*) Note that in *Classifiers only*, it is only evaluated whether the particular cell is dividing whereas in the tracking models, we go beyond that and additionally require the correct links to the daughter cells. The ground truth of dataset B (dataset C) contains 56,029 (34,985) moves, 216 (440) divisions, 1,878 (1,189) mergers, and 1,466 (533) resolved mergers events.



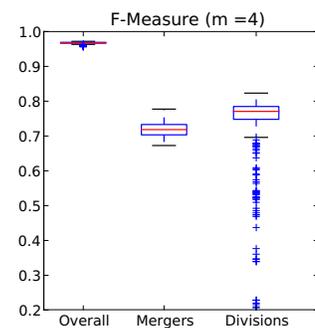
(a) 2D projection of the 3D trajectories of dataset B (*Drosophila* 3D+t)



(b) One frame of dataset C (Mitocheck 2D+t)



(c) Space-time rendering of the trajectories of dataset C



(d) Parameter sensitivity

Figure 6: (a) 2D projection of the 3D trajectories of dataset B (*Drosophila*, 3D+t) over all 100 time steps. (b) One frame of dataset C (2D+t) and (c) the 3D space-time rendering of its trajectories. Note that daughter cells inherit the color of their mother cell. (d) F-measures for dataset C for a search over 720 parameter configurations.

identities of such merged objects. In particular, the associations between the distinct objects after demerging are evaluated as true positives only if they link to the true respective objects before merging – possibly over long sequences of being merged.

Dataset C is a publicly available 2D+t dataset taken from the Mitocheck project² (92 time steps, $1,344 \times 1,024$ pixels), segmented using the method in [9] and a gold standard is acquired manually. In our model, we again treat each connected component as one detection and set the parameters (for $m = 4$) to $\alpha = 5$, $w_{\text{app}} = 100$, $w_{\text{van}} = 100$, $w_{\text{tr}} = 10$, $w_{\text{div}} = 16$. Due to the global mass conservation, our model (f-measure of 0.80 and 0.76 for divisions and mergers) improves significantly over the results of the rather weak local division and merger classifiers (0.70 and 0.49, respectively), cf. Tab. 2. In Fig. 6d the robustness of the model parameters is addressed for this experiment in the case of $m = 4$ for a search over a reasonable parameter range (720 evaluations). The influence of the parameter setting on the overall result is marginal due to the domination of move events, which are robust to parameters. The results of mergers and divisions seem to depend more on the parameter setting, however, the standard deviation is only 0.02 and 0.08, respectively.

The results of our model can be further improved by designing even more features for object classification and division detection. This additional local evidence can then be put into global context within the factor graph. It should be noted that the object classification and division detection modules can be fully adopted to the particular application domain.

5. Conclusion

We have proposed a probabilistic graphical model which – due to the explicit modeling of global conservation laws – can robustly correct errors from a previous detection step. We have shown that the proposed factor graph can outperform a recently published cell tracking method on sequences of proliferating cells in a dense populations thanks to the consideration of over- and undersegmentation errors. In addition, our model can partition and track previously merged objects while preserving their original identities. In future work, we would like to extend our model to address the only remaining segmentation error, viz. undetected objects. Furthermore, we plan to learn the parameters from (sparse) user annotations.

Acknowledgements We are very grateful for partial financial support of the CellNetworks Cluster and want to thank Jean-Karim Heriche (EMBL Heidelberg) for providing the Mitocheck dataset.

²www.mitocheck.org/archive/cgi-bin/mtc?action=show_movie;query=87214

References

- [1] H. Ben Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. *ICCV*, 2011. 3
- [2] B. Bose, X. Wang, and E. Grimson. Multi-class object tracking algorithm that handles fragmentation and grouping. In *CVPR*, 2007. 3
- [3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 4, 6
- [4] E. Fox, D. Choi, and A. Willsky. Nonparametric Bayesian methods for large scale multi-target tracking. In *ACSSC*, 2006. 3
- [5] B. X. Kausler, M. Schiegg, B. Andres, M. Lindner, U. Koethe, H. Leitte, J. Wittbrodt, L. Hufnagel, F. A. Hamprecht, and U. Köthe. A Discrete Chain Graph Model for 3d+t Cell Tracking with High Mis-detection Robustness. In *ECCV*, 2012. 3, 6, 7
- [6] D. Koller and N. Friedman. *Probabilistic Graphical Models*, 2010. 3
- [7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001. 3
- [8] X. Lou and F. A. Hamprecht. Structured Learning for Cell Tracking. *NIPS*, 2011. 3
- [9] X. Lou, U. Koethe, J. Wittbrodt, and F. A. Hamprecht. Learning to Segment Dense Cell Nuclei with Shape Prior. In *CVPR*, 2012. 8
- [10] E. Meijering, O. Dzyubachyk, and I. Smal. Methods for cell and particle tracking. *Methods in Enzymology: Live Cell Imaging*, 2012. 1
- [11] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking: Linking identities using Bayesian network inference. In *CVPR*, 2006. 3
- [12] D. R. Padfield, J. Rittscher, and B. Roysam. Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis. *Medical Image Analysis*, 15(4):650–668, 2011. 3
- [13] J. C. Rubio, J. Serrat, A. M. Lopez, and D. Ponsa. Multiple target tracking for intelligent headlights control. *13th International IEEE Conference on Intelligent Transportation Systems*, pages 903–910, Sept. 2010. 1
- [14] S. Sahni. Computationally related problems. *SIAM Journal on Computing*, 3(4):262–279, 1974. 2, 3
- [15] C. Sommer, C. Straehle, U. Kothe, and F. A. Hamprecht. Ilastik: Interactive learning and segmentation toolkit. *ISBI*, (1):230–233, 2011. 6
- [16] C. Wang, M. de La Gorce, and N. Paragios. Segmentation, ordering and multi-object tracking using graphical models. In *ICCV*, 2009. 2
- [17] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *CVPR*, 2012. 2
- [18] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys (CSUR)*, 38(4), 2006. 1
- [19] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008. 3