Doctoral Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Probabilistic Models for Joint Segmentation, Detection and Tracking

Tomáš Sixta

Supervisor: doc. Boris Flach, Dr. rer. nat. habil.
Ph.D. programme: Electrical Engineering and Information Technology
Field of study: Artificial Intelligence and Biocybernetics
August 2018

Acknowledgements

I would like to express my gratitude to my advisor doc. Boris Flach, Dr. rer. nat. habil. for his valuable suggestions and honest feedback. His guidance allowed me to learn from my mistakes and become a better researcher and stronger person.

I was privileged to be a research intern at the Machine Learning Research Group at the University of Guelph. I would like to thank to the group leader, Graham Taylor, for making my internship happen and Daniel Jiwoong Im and Ethan Buchman for inspiring discussions and fruitful cooperation. My stay was indeed a great experience and opportunity.

I would like to thank to my colleagues and friends from the Center for Machine Perception for countless research related discussions as well as friendly conversations. They helped to create a truly pleasant working environment.

I would also like to thank to my parents and friends, who supported me selflessly regardless how powerless I felt. Without their help I would never be able to finish this thesis.

Finally, I gratefully acknowledge that my research was supported by the Grant Agency of the Czech Republic projects P202/12/2071 and 16-05872S and the Grant Agency of the CTU projects SGS11/130/OHK3/2T/13 and SGS15/156/OHK3/2T/13.

Declaration

I hereby declare that I have completed this thesis independently and that I have listed all used information sources in accordance with Methodical instruction about ethical principles in the preparation of university theses.

In Prague, 22. August 2018

Abstract

Migration of cells and subcellular particles plays a crucial role in many processes in living organisms. Despite its importance a systematic research of cell motility has only been possible in last two decades due to rapid development of non-invasive imaging techniques and digital cameras. Modern imaging systems allow to study large populations with thousands of cells. Manual analysis of the acquired data is infeasible, because in order to gain insight into underlying biochemical processes it is sometimes necessary to determine shape, velocity and other characteristics of individual cells. Thus there is a high demand for automatic methods.

Lots of methods for multiobject tracking consist of two steps: detection of individual cells in all frames and linking the detection candidates into a final lineage. This approach (tracking-by-detection) allows to apply single tracking method to multiple domains simply by changing the detector but at the same time suffers from propagation of detection errors.

The aim of this thesis is to create a method for segmentation, detection and tracking of multiple objects that integrates the detection and linking steps into one joint model in order to eliminate the error propagation problem. As an initial step towards this goal we propose a novel method for tracking dimensionless particles. Unlike traditional tracking-by-detection approaches it uses a map of detection scores instead of a fixed set of detection candidates which allows the tracker to partially overcome some detection errors. Although this method integrates the detection and linking steps only partially, experimental evaluation on a publicly available dataset suggests that it is competitive with state-of-the-art methods.

Our second contribution is a shape prior for cell segmentation based on Markov Random Fields on bipartite graphs. The model uses multilabel hidden variables to model middle-level shape characteristics, e.g. smoothness of the boundary or sharp tips. We also propose a novel algorithm for unsupervised parameter learning suitable for this class of models.

The main contribution of this thesis is a novel method for joint segmentation, detection and tracking of multiple objects. The method is based on a probabilistic model that is defined implicitly in terms of a Markov chain Monte Carlo (MCMC) algorithm. It contains a temporal feedback, which allows to dynamically alter detector parameters using hints given by neighboring frames and, in this way, correct detection errors. The parameters of the model are learned using an objective based on empirical risk minimization. The performance of the method is tested on various datasets.

Keywords: Tracking; Detection; Segmentation; Graphical Models; Parameter Learning

Supervisor: doc. Boris Flach, Dr. rer. nat. habil.

Abstrakt

Migrace buněk a buněčných částic hraje důležitou roli ve fungování živých organismů. Systematický výzkum buněčné migrace byl umožněn v posledních dvaceti letech rychlým rozvojem neinvazivních zobrazovacích technik a digitálních snímačů. Moderní zobrazovací systémy dovolují studovat chování buněčných populací složených z mnoha ticíců buněk. Manuální analýza takového množství dat by byla velice zdlouhavá, protože některé experimenty vyžadují analyzovat tvar, rychlost a další charakteristiky jednotlivých buněk. Z tohoto důvodu je ve vědecké komunitě velká poptávka po automatických metodách.

Velké množství metod pro sledování pohybu více objektů pracuje ve dvou fázích: detekce jednotlivých buněk ve všech snímcích videa a spojování těchto detekčních kandidátů do výsledných trajektorií. Tento přístup (tracking-by-detection) dovoluje kombinovat jeden sledovací algoritmus s různými detektory a díky tomu jej aplikovat na různé druhy dat. Zároveň však neumožňuje opravit detekční chyby, které negativně ovlivňují kvalitu výsledných trajektorií.

Cílem této práce je vytvořit metodu pro segmentaci, detekci a sledování více objektů, která spojuje detekci a sledování do jednom modelu a tím potlačuje citlivost na chyby detektoru. Prvním krokem k tomuto cíli je nová metoda pro sledování bezrozměrných objektů. Oproti tradičním postupům, které používají neměnnou množinu detekčních kandidátů, využívá tato metoda mapu detekčních skóre, čímž umožňuje sledovacímu algoritmu opravit některé detekční chyby. Přestože tato metoda spojuje detekci a sledování jen v omezené míře, výsledky experimentů naznačují, že je plně srovnatelná s konkurenčními postupy.

Druhým příspěvkem této práce je apriorní model tvaru pro segmentaci buněk, který je založen na markovských náhodných polích na bipartitních grafech. Tento model využívá skryté proměnné s více než dvěma možnými značkami a umožňuje modelovat lokálně-globální charakteristiky tvaru, např. hladkost hranice buňky či ostré výběžky. Dále je navržen nový algoritmus pro učení parametrů bez učitele, který je využitelný pro tuto třídu modelů.

Hlavním příspěvkem této práce je nová metoda pro sdruženou segmentaci, detekci a sledování více objektů. Metoda je založena na pravděpodobnostním modelu, který je definován implicitně pomocí algoritmu Markov Chain Monte Carlo (MCMC). Tento algoritmus využívá zpětnovazební mechanismus, který umožňuje sledovacímu algoritmu dynamicky měnit parametry detektoru na základě vlastností objektů v okolních snímcích videa a tímto způsobem potlačit citlivost na detekční chyby. Parametry modelu jsou učeny pomocí kritéria odvozeného z minimalizace empirického rizika. Metoda je otestována na různých typech dat.

Klíčová slova: Sledování; Detekce; Segmentace; Grafické modely; Učení parametrů

Contents

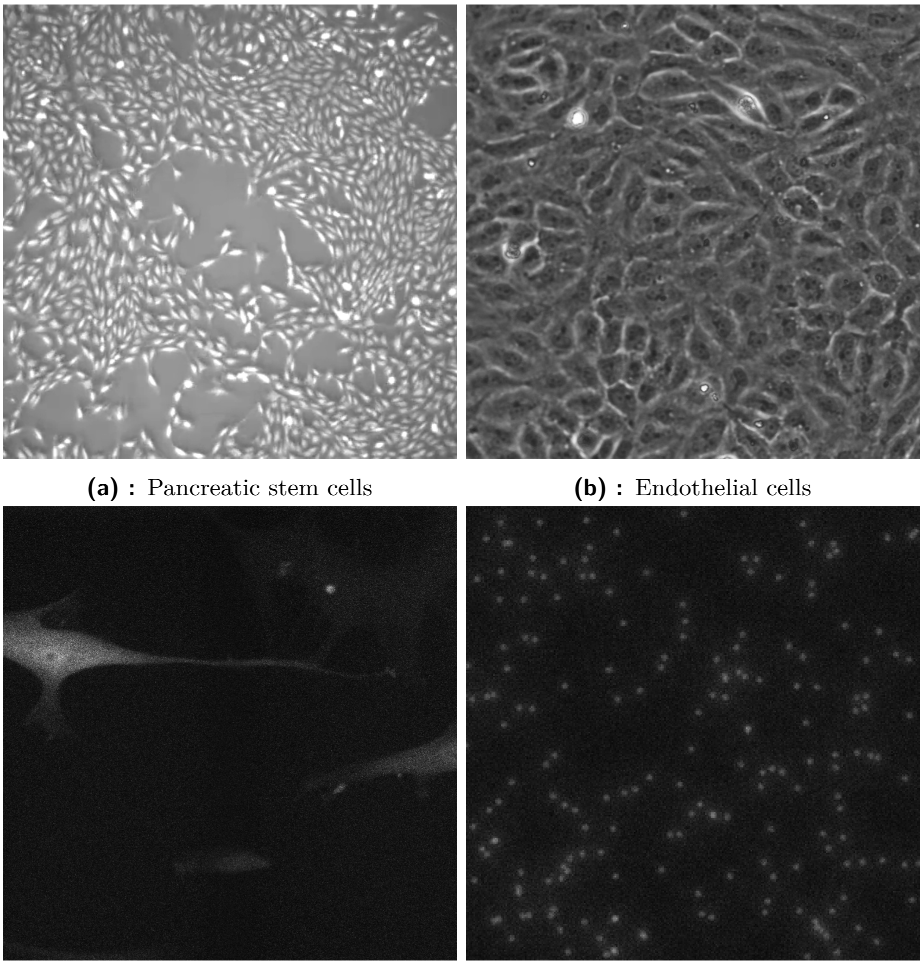
Chapter 1

Introduction

Cell migration is a fundamental process in multicellular organisms [192]. It plays a crucial role in the embryonic development, wound healing, immune responses and many other processes essential for the development and maintenance of the organism. It is also a key factor of various diseases: invasion of tumor cells into adjacent tissues, atherosclerosis, developmental brain malformations caused by a neuronal migration disorder, etc [62]. Understanding the underlying biochemical principles of cell motility may lead to the development of new ways of treating such diseases. Thus, cell migration is a central topic of research in biology and medicine.

Traditional approaches for microscopy imaging do not allow to study migrating cells. As most living cells are translucent they cannot be observed directly by a traditional light microscope, but it is necessary to stain them first in order to enhance contrast of the recorded images [97]. Staining is an inherently invasive procedure, which affects or even disrupts normal cell behavior and consequently the recorded data have only a limited value for understanding the dynamics of the cell populations. Systematic research of cell motility has only been possible in last two decades due to wider availability of non-invasive imaging techniques (e.g. phase contrast microscopy) and high resolution digital cameras [186]. Modern imaging systems for time-lapse microscopy are in addition capable of maintaining specified temperature, humidity, concentration of carbon dioxide and other environmental parameters and allow to observe migrating cells in close to in-vivo conditions [39, 201].

Analysis of the recorded data is as important as their acquisition. In order to gain insight into biochemical processes regulating the cell movement it is sometimes necessary to determine shape, velocity and other characteristics of individual cells [184], which makes the manual analysis infeasible, because modern imaging systems allow to record videos of thousands of cells over hundreds of time steps. Semi automatic methods require to annotate the cells manually in the initial frame and then track them till the end of the video sequence. Depending on the method the user can also interactively alter the tracking results in order to correct obvious errors [107, 146, 190]. Although semi automatic approaches make the analysis less labor intensive, they are still susceptible to fatigue-related errors and suffer from low repeatability due to disagreements between different users (researchers) [63]. Consequently



(a) : Pancreatic stem cells

(b) : Endothelial cells

(c) : Rat mesenchymal stem cells

(d) : Vesicles in the cytoplasm

Figure 1.1: Sample biomedical images.

there is a high demand for fully automatic methods.

Despite rapid development, methods for automatic segmentation, detection and tracking of migrating cells are still in their infancy. They typically consist of two separate steps. In the first step (detection) they try to locate individual objects and estimate their shapes and in the following step (linking) they link the resulting detection candidates into the final lineage [116, 125]. This approach is commonly referred to as tracking-by-detection and although it enjoys many advantages (e.g. it allows to apply the method to various domains simply by changing the detector in the first step), the quality of the tracking results is negatively influenced by propagation of detection errors into the linking step. This problem affects the analysis of cell motility severely – even modern systems for time-lapse microscopy produce images with low signal-to-noise ratio [89], poor visibility of cell boundaries in densely populated areas and other ambiguities that cannot be resolved by the detector alone (Figure 1.1). Although the problem of error propagation can be partially mitigated by generating more detection candidates than objects and giving the linking step freedom to choose the optimal subset [3, 58, 195], in its full generality it remains largely unsolved.

1.1 Problem Formulation

In this section we formally define the problem which we tackle in this thesis, establish basic terminology and outline our goals we wish to fulfill. The motivation for this thesis is analysis of time-lapse microscopy videos. The analysis typically consists of three steps. In the first step a cell culture is cultivated and put under a microscope which records its behavior for several hours or days, depending on properties of the studied phenomenon. In the second step the acquired data (a sequence of images) are processed in order to estimate poses of individual cells in every frame of the video, their lineage and detect certain events occurring in the population, e.g. cell divisions or apoptosis. Finally, the results of the second step are used to calculate quantities relevant for the studied phenomenon.

In this thesis we focus on tasks related to the second step – segmentation, detection and tracking. As there is no widely accepted definition of these tasks, we define them for the purpose of this thesis as follows:

- **Segmentation** is a labelling of pixels of the input sequence which encodes a "class" of each pixel, e.g. background, cell nucleus, cytoplasm, etc. This is commonly referred to as "semantic segmentation".

- **Detection** is a process of estimating poses of individual objects in each image. The definition of an object depends on the underlying biological motivation for the analysis: they might be the cells, their parts (e.g. nuclei, organelles, etc.) or even certain spatially localized events (cell division, apoptosis, phagocytosis, etc.). Pose is any information, that allows to localize an object in the image, e.g. the centroid, scale, bounding box, etc. In the tracking-by-detection paradigm the pose is estimated by a detector, which may however generate several options for some objects or suffer from false positive errors. To prevent confusion we refer to an output of the detector that *may* correspond to a single object in a single frame as a detection candidate. Formally we define the pose of an object as a labelling of pixels of the image, which allows for each pixel to determine, whether it belongs to the object or not. In the literature this is referred to as "instance segmentation". To avoid confusion, in the rest of this thesis we will use word segmentation solely in the meaning of semantic segmentation and denote instance segmentation as detection.

- **Tracking** is a process of estimating trajectories of moving objects. In the tracking-by-detection paradigm this means linking detection candidates into the final tracking result. For the purpose of this thesis we define a tracking result as an information that allows to reconstruct the trajectories and shapes of individual objects as well as their complete lineage.

Our goal is to create a probabilistic model for joint segmentation, detection and tracking. Solving all three subtasks in one joint model is attractive for both practical and theoretical reasons, because it eliminates the error propagation problem. In the same time the model should retain the flexibility of the tracking-by-detection methods, i.e., it should be possible to apply it to multiple domains simply by changing its detection-related component. Special attention should be given to parameter learning (in terms of a learning objective and an algorithm), because automatic parameter learning is another important factor that allows to apply the model to various domains. In addition it is also our goal to propose an inference procedure that will allow to use the model as a basis of a practical method for detection and tracking of several thousands cells in time-lapse microscopy videos with hundreds of frames.

As a secondary goal we wish to create a probabilistic shape prior for cell segmentation. Although living cells often do not have a "characteristic" global shape, their boundaries tend to be locally smooth. Therefore we are interested in priors that can model local properties of the boundary because such a model could improve segmentation accuracy for wide range of types of cells.

1.2 Thesis Organization

In this thesis we present the steps taken in the effort to fulfill the aforementioned goals. In chapter 2 we summarize the theory and state-of-the-art methods related to this thesis. The chapter begins with a brief introduction to probabilistic graphical models which are in the core of our contributions. We establish the basic notation and terminology, formalize parameter learning and highlight main principles of the inference. Furthermore, we summarize the state-of-the-art methods used for segmentation, detection and tracking multiple objects. Since this thesis is motivated by analysis of microscopy videos of living cells, we focus on methods suitable for biomedical data.

In chapter 3 we present a novel method for tracking dimensionless particles. This method can be seen as a first step taken towards the main goal of this thesis. It is based on a probabilistic graphical model for sets of moving objects, each represented by its trajectory, and it allows to model mutual interactions among the objects. The inference task is defined as a minimization of Bayes risk and the performance of the method is evaluated on a publicly available dataset.

In chapter 4 we present a probabilistic shape model for cell segmentation. The model is a Markov Random Field (MRF) on a bipartite graph. Variables of one layer encode a segmentation of the corresponding pixels and the variables in the second layer serve as a regularizer for the segmentation within their receptive fields. Furthermore, we propose a novel method for unsupervised parameter learning for this class of models.

In chapter 5 we propose a novel method for joint segmentation, detection and tracking. The method is based on a probabilistic model that is defined implicitly in terms of a Markov chain Monte Carlo (MCMC) algorithm. It contains a temporal feedback, which allows to dynamically alter detector parameters using hints given by neighboring frames and, in this way, correct detection errors. The parameters of the model are learned using an objective based on empirical risk minimization. The performance of the method is tested on various datasets.

1.3 Contributions

The main contributions of this thesis as follows:

- We developed a probabilistic graphical model for tracking dimensionless objects. It allows to model the object appearance, motion and their mutual interactions jointly and it can be easily extended by adding new types of potential functions. Furthermore it does not expect a fixed set of detection candidates as an input, but instead uses a probabilistic detection map in order to infer the number of objects and their positions.

- We proposed a probabilistic shape prior for semantic segmentation, which allows to model middle-level shape properties, e.g. smooth boundaries. The model is a MRF on bipartite graph. Unlike most similar models found in the literature, it uses multilabel variables to represent different shape characteristics instead of large number of binary variables.

- We proposed a novel algorithm for unsupervised parameter learning of MRFs on bipartite graphs. The method is a modified EM algorithm, which in the M step uses the maximum pseudolikelihood estimator instead of the likelihood.

- We created a probabilistic model for tracking multiple objects, which integrates a tracker and an arbitrary detector into one joint model. This is achieved by a feedback mechanism, which allows to revisit shapes and positions of detection candidates based on poses of objects in nearby frames. The model is defined implicitly in terms of a MCMC algorithm, which builds on advantages of graphical models.

- We formulated a novel parameter learning objective for probabilistic models which are defined implicitly in terms of a MCMC algorithm. The objective is derived such that the MCMC algorithm converges quickly with high probability to a state similar (in terms of a loss function) to a training sample.

Chapter 2

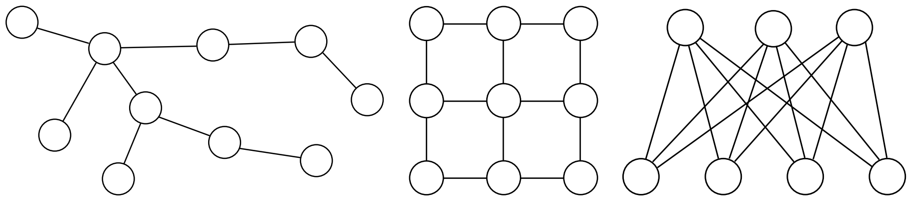
Related Work

In this chapter we summarize the theory and state-of-the-art methods related to this thesis. Since the probabilistic graphical models are the core of our contributions described in chapters 3 - 5, we begin with their brief introduction. Furthermore, we summarize the state-of-the-art methods for segmentation, detection and tracking of multiple objects and highlight their strengths and weaknesses. Since these tasks are fundamental problems of computer vision it is beyond the scope of this thesis to provide an exhaustive overview. Instead, we primarily focus on methods applicable to biomedical images.

2.1 Probabilistic Graphical Models

One of the primary goals of artificial intelligence is to create an automatic system capable of understanding content of digital images. Despite a tremendous effort invested in computer vision in the last decades this problem is still far from being solved. One of the reasons why an universal method still does not exist is the uncertainty inherent to all stages of image understanding. An early source of uncertainty is the data acquisition. Even high quality sensors are up to some extent noisy and as a result there is an uncertainty about the true intensities of the image pixels. Although the characteristics of the noise can be in principle deduced e.g. from the temperature of the sensor or type of illumination, such information is rarely available in practice and therefore the uncertainty must be dealt with in a different way. As another example, consider a sequence of time-lapse microscopy images of migrating cells. The appearance, positions and shapes of individual cells change across the frames and as a result it is not possible without a genetic analysis to exactly determine, whether two detection candidates correspond to a single physical cell, or to pinpoint the exact moment of cell division. Furthermore in some modalities it may not be possible to identify individual cells in a confluent culture, because the imaging technique is unable to visualize the cell membranes.

Although the aforementioned uncertainties have different causes, it is convenient to model them in an unified framework. A natural choice is the probability theory, specifically graphical models, which allow to model uncertainty in systems composed of many interacting parts.



(a) : Tree **(b) :** Lattice **(c) :** Bipartite graph

Figure 2.1: Typical structures of Markov Random Fields

Graphical models are probabilistic distributions over sets of random variables $X = \{X_i \mid i = 1..n\}$, which represent the conditional independence relationships between the variables by an underlying graph $G = (V, E)$ such that its nodes $V_i \in V$ correspond to the random variables. The variables can be discrete, continuous or mixed. The edges link together pairs of variables but the theory of graphical models allows to incorporate hyperedges as well and, in this way, to introduce higher order dependencies into the model.

There are two main families of graphical models distinguished by the type of edges in the underlying graph: directed (Bayesian networks) and undirected (Markov Random Fields). Bayesian networks allow to model causal relationships between random variables, which makes them convenient for many tasks of artificial intelligence, including semantic search, knowledge representation or diagnostic reasoning. Markov Random Fields, on the other hand, can naturally model soft constraints between random variables, which makes them more suitable (and consequently popular) for low level computer vision. Since Bayesian networks are rarely used for segmentation or detection, we focus in the rest of this section to Markov Random Fields only.

2.1.1 Markov Random Fields

Markov Random Fields (MRF) are undirected graphical models, which represent the conditional independence relationships explicitly by an underlying graph $G = (V, E)$. Two sets of variables $X_A \subset X$, $X_B \subset X$ are conditionally independent given a set $X_S \subset X$

$$p(x_A, x_B \mid x_S) = p(x_A \mid x_S)p(x_B \mid x_S) \tag{2.1}$$

if all paths between the corresponding sets of nodes V_A and V_B are blocked by (have to pass through) nodes in V_S.

The Hammersley-Clifford theorem [72] states that strictly positive MRFs (i.e., $p(x) > 0$ for all realizations of X) can be equivalently parameterized in terms of potential functions associated with the (maximal) cliques of the underlying graph:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp \sum_{c \in E} \psi_{\boldsymbol{\theta}_c}(x_c), \tag{2.2}$$

where θ denotes the set of all parameters, $\boldsymbol{\theta}_c$ is the parameter vector associated with clique c and

$$Z(\theta) = \sum_x \exp \sum_{c \in E} \psi_{\boldsymbol{\theta}_c}(x_c) \tag{2.3}$$

is the partition function. Common structures of the underlying graph are e.g. trees, lattices or bipartite graphs (Figure 2.1).

In practice, the potential functions $\psi_{\boldsymbol{\theta}_c}$ are often specified in "tabular" form, i.e., every configuration (labelling) of x_c is assigned a fixed value. In this case the distribution function (2.2) can be equivalently written using a vector valued indicator function ζ, which sets the component of the output vector corresponding to the labelling of x_c to 1 and the other components to 0:

$$p_\theta(x) = \frac{1}{Z(\theta)} \exp \sum_{c \in E} \langle \boldsymbol{\zeta}(x_c), \boldsymbol{\theta}_c \rangle, \tag{2.4}$$

These models are also called log-linear, because the argument of exp is a linear combination of the parameters.

In computer vision, the MRFs were traditionally used as joint models of images and their segmentations, i.e., $p_\theta(S, I) = p_\theta(S) p_\theta(I \mid S)$, where I denotes an image and S the segmentation. These models allow to estimate a segmentation given an image but at the same time are able to generate a random image given a segmentation. Thus, they are called generative models. In order to keep the model tractable, the conditional distribution $p_\theta(I \mid S)$ must in practice adopt many simplifying assumptions (e.g. conditional independence of pixels), which may be too restrictive for the task in mind. An alternative approach is to use an MRF to model $p_\theta(S \mid I)$ directly. This class of models is called Conditional Random Fields (CRF) [108]. CRFs make no assumption about the prior distribution of images $p_\theta(I)$, which allows to use more complex models without loosing tractability. However, since they are discriminative models, they have no way of sampling random images and do not allow unsupervised learning.

2.1.2 Parameter Learning

Even moderately sized MRFs have large number of parameters, which makes manual parameter tuning infeasible. Therefore in practice the parameters are learned by an automatic method from the training data. Formally, we assume that we are given a set of i.i.d. (independent and identically distributed) training labellings $\mathcal{T} = \{x^1, x^2, ..., x^n\}$ and we seek the parameters θ^*, which maximize some objective function.

A natural choice for the objective function is the log-likelihood:

$$\begin{aligned} L(\theta) &= \log \prod_{a=1}^{n} p_\theta(x^a) \\ &= \left(\sum_{a=1}^{n} \sum_{c \in E} \psi_{\boldsymbol{\theta}_c}(x_c^a) - n \log Z(\theta) \right). \end{aligned} \tag{2.5}$$

It can be shown, that for log-linear models the likelihood (2.5) is a concave function. If we for convenience represent the parameters in the exponential form $\theta_{x_c} = \exp \psi_{\boldsymbol{\theta}_c}(x_c)$ the gradient can be written as

$$\frac{\partial L(\theta)}{\partial \theta_{x_c}} = \frac{\sum_{a=1}^{n} \delta(x_c, x_c^a)}{\theta_{x_c}} - n \frac{p_\theta(x_c)}{\theta_{x_c}}, \tag{2.6}$$

where δ is the Kronecker's delta. This expression provides an intuitive interpretation of the resulting optimization task: the optimal parameters θ^* given by the maximum likelihood (ML) estimator are such that the marginal probability $p_\theta(x_c)$ coincides with the frequency of x_c in the training data. Calculating the marginals is however #P-hard [27, 40], which means, that the gradient cannot be in general calculated exactly and the optimal parameters can be found only approximately, e.g. by stochastic gradient ascent.

Common approach, how to deal with the computational complexity of ML learning, is to replace the "full" likelihood by a related objective function, which is easier to optimize. A large family of these objectives are the composite likelihoods (CL) [113, 197]. The most widely used CL estimator is the pseudolikelihood [19]. It approximates the joint distribution by product of conditional marginals of single variables given their neighbors, i.e.,

$$p_\theta(x) \approx \prod_{i \in V} p_\theta(x_i | x_{\mathcal{N}(i)}). \tag{2.7}$$

Similarly to the likelihood, the pseudolikelihood (2.7) is also a concave function. It is also a consistent estimator, although it has higher variance than the likelihood.

Minimum probability flow (MPF) [180] is an example of a learning objective, which was not designed as a mere approximation of the likelihood. It defines the learning objective as a flow of probability from states corresponding to the training data to other states. The flow dynamics are defined such that minimizing the flow is equivalent to minimizing the KL divergence between the model and the empirical distribution. The MPF is a consistent estimator and in addition it is convex for models in the exponential family (which includes MRFs).

Unsupervised Learning

So far we assumed, that the training samples are specified for all variables of the model. However, in practice the training data are sometimes incomplete. Consider for example a segmentation model $p_\theta(S, I)$ for cells in microscopy images. Acquiring an image of cell culture is relatively straightforward but generating its annotation is labor intensive and consequently it might be necessary to learn the parameters of the model from the images only. Variables, for which training data are available, are called visible whereas those with no data are hidden or latent. In this scenario a learning objective can be obtained by marginalizing over the latent variables, which in case of maximum likelihood estimator yields

$$\theta^* = \operatorname*{argmax}_\theta \log \prod_{a=1}^{n} \sum_{S} p_\theta(S, I). \tag{2.8}$$

This optimization task is no longer concave and therefore a gradient ascent algorithm can get stuck in a local optimum. There are several methods for unsupervised learning of MRFs, e.g. Baum-Welch algorithm for MRFs on trees [14], EM algorithm [45, 169] or persistent contrastive divergence [188, 189]. We discuss some options in more detail in chapter 4 where we also propose a new method for unsupervised parameter learning.

2.1.3 Inference

Problems of computer vision are typically formulated as optimization tasks, i.e., finding a solution, which is optimal with respect to the model of the problem. For example the segmentation problem can be formulated as follows: given an image I, find its segmentation S such that it is optimal with respect to $p_\theta(S, I)$. Methods based on probabilistic models typically seek a solution with maximum a posteriori probability

$$S^*_{MAP} = \operatorname*{argmax}_S p_\theta(S \mid I). \tag{2.9}$$

This choice is intuitive and simple in the sense that the quality of a solution is determined directly by the model. However in practice the MAP solution is not always desirable, e.g. when the MAP solution is not feasible in real world.

In methods proposed in this thesis we define the inference task as minimization of Bayes risk, i.e., the posterior expected value of a loss function. Using notation from the segmentation example, this results in the following optimization task:

$$S^* = \operatorname*{argmin}_{\hat{S}} \sum_S L(S, \hat{S}) p_\theta(S \mid I). \tag{2.10}$$

The loss function L can be used to quantify the cost of different types of errors. For example, consider segmentation of cells in a microscopy image into background, cytoplasm and nucleus and suppose that the evaluation of the underlying biological experiment requires to accurately localize cell nuclei. Making an error involving the nucleus label is then more costly than misclassifying a cytoplasm or a background pixel, which can be reflected in the loss function. There are many types of losses used in the literature. In the remainder of this subsection we briefly describe the most widely used types: the 0-1 loss and the additive loss.

0-1 Loss

The 0-1 loss considers all types of errors equally costly:

$$L_{01}(S, \hat{S}) = \begin{cases} 0 & S = \hat{S} \\ 1 & \text{else,} \end{cases} \tag{2.11}$$

It is easy to show that the 0-1 loss leads to MAP estimation. This makes it (implicitly) very popular in computer vision, because most methods based on probabilistic graphical models define their inference task as MAP estimation without considering possible alternatives.

MAP estimation of MRFs is an NP-hard problem [176] which can be in general solved only approximately, e.g. by Mean Field algorithm [64], Loopy Belief Propagation [141], Iterated Conditional Modes algorithm [20] or Tree-Reweighted Message Passing [199, 200]. However, for certain types of MRFs the MAP estimate can be found exactly in polynomial time. In the context of low-level vision a particularly important class are binary pairwise MRFs with submodular potentials, i.e.,

$$\forall ij \in E : \psi_{\boldsymbol{\theta}_{ij}}(B, B) + \psi_{\boldsymbol{\theta}_{ij}}(F, F) \leq \psi_{\boldsymbol{\theta}_{ij}}(B, F) + \psi_{\boldsymbol{\theta}_{ij}}(F, B), \tag{2.12}$$

where B and F represent possible labels of individual variables. The MAP estimation of these models can be formulated as max-flow/min-cut problem and solved by a polynomial algorithm [24]. Up to some extent this can be generalized to multilabel pairwise MRFs. If every potential function $\psi_{\boldsymbol{\theta}_{ij}}$ is a metric, the α-expansion algorithm [25] finds a local optimum of 2.9 within a known factor of the global optimum.

Additive Loss

The additive loss function defines the cost of an error by the number of misclassified pixels:

$$L_{add}(S, \hat{S}) = \sum_i \|S_i - \hat{S}_i\|, \tag{2.13}$$

where S_i is a label of pixel i. The optimal labelling S^* can be obtained for each pixel separately by taking a label with maximum marginal probability:

$$S_i^* = \operatorname*{argmax}_{S_i} p_\theta(S_i \mid I). \tag{2.14}$$

Calculation of Marginal Probabilities

Marginal probabilities needed for the inference with the additive loss can be calculated by the Junction tree (JT) algorithm, which decomposes the original graph into a junction tree and uses belief propagation to calculate its marginals. The time complexity of the JT algorithm is exponential with respect to the treewidth of the graph, i.e., the size of the largest vertex set in its tree decomposition, which makes it practical only for a limited class of models with low treewidth. Marginal probabilities can be also calculated exactly in polynomial time for MRFs on complete graphs with homogeneous pairwise potentials [60]. However since calculating the marginals is a #P-hard problem, there is no exact polynomial method for MRFs with unrestricted structure and consequently the inference task can be in general solved only approximately.

Approximate methods for calculating the marginals can be roughly categorized into two groups: stochastic and deterministic. Stochastic methods approximate the marginals by averaging over the samples drawn from the model. Since generating exact samples from a MRF is in general infeasible, they are typically produced by a Markov Chain Monte Carlo method, e.g. by the Gibbs sampler [65]. There is a huge number of sampling methods that aim at improving the properties of the Gibbs sampler, especially its mixing rate (i.e., the speed of convergence to the model distribution). In the context of this thesis, a particularly important method is the blocked Gibbs sampler, which in each iteration updates multiple variables (as opposed to the Gibbs sampler, which resamples one variable at a time). This is especially beneficial for MRFs on bipartite graphs. As variables of one layer are conditionally independent given the variables of the other layer, the whole model can be updated in only two iterations.

Lots of deterministic methods for approximate MAP inference can be also used for estimating the marginals. Belief Propagation (BP) [141] estimates the marginals by iteratively sending messages along edges of the underlying graph. The algorithm is exact for trees and converges in linear time with respect to the number of variables. It can be also used for general graphs (it is then called the Loopy Belief Propagation), but although it often provides a "good" estimate of the true marginals, it gives no indication how accurate the estimate is and in addition the algorithm is even not guaranteed to converge. Ihler et al. [85] derived convergence conditions and strict bounds and estimates of the resulting error of the BP algorithm. Yedidia et al. [211] related BP to the minimization of so called Bethe free energy, which allowed development of methods that minimize it directly and always converge [214].

BP belongs to a large family of algorithms called variational methods [211]. They project the original MRF into a simpler model, where exact inference is feasible and which most closely resembles the original model. Another example of variational methods is the Mean Field algorithm [64]. It approximates the original distribution $p_\theta(x)$ by another distribution $q(x)$ with all the variables independent such that the Kullback-Leibler divergence between q and p_θ

$$KL(p_\theta||q) = \sum_x p_\theta(x) \log \frac{p_\theta(x)}{q(x)} \tag{2.15}$$

is minimized. Bounded Treewidth Subgraph algorithm [59] approximates the original distribution by a model on a graph with low treewidth and performs the inference in this submodel. Tree-Reweighted Message Passing [199, 200], and its variants (e.g. [103]) approximates the original distribution by convex combination of tree-structured distributions which is used as an input for a message passing algorithm.

2.2 Segmentation & Detection

2.2.1 Thresholding

Historically the first but still widely used method for biomedical image segmentation is intensity thresholding [128]. It is based on assumption that the objects of interest appear as bright homogeneous regions on darker background. The thresholding labels every pixel i of the gray-scale image I

either as background (B) or foreground (F):

$$S_i = \begin{cases} F & I_i \geq \lambda_i \\ B & \text{else,} \end{cases} \tag{2.16}$$

where λ_i is the threshold. Methods that utilize a single threshold for all pixels are called global, whereas approaches, that allow to choose the threshold for each pixel separately, are referred to as adaptive thresholding methods [204].

Since thresholding alone can provide semantic segmentation only, it must be followed by an additional step that splits the foreground cluster into individual objects. The simplest option is to consider each connected component of the foreground as an object [204]. More elaborate methods consider the connected components only as detection candidates and the final decision, whether they correspond to an object or the background, is made based on their geometrical properties and appearance of the underlying region in the original image [9]. This allows to discard false-positive regions created as a result of noise in the input image. Other methods employ mathematical morphology in order to remove small false-positive regions, separate touching objects and/or merge foreground regions, that correspond to a single object [48, 81, 90].

Global Thresholding

Global thresholding methods select a single threshold λ_i^* by analyzing the histogram of the whole image. Since there is enormous number of methods in the literature, we describe in this section only the most popular approaches and refer the reader to [174] for more comprehensive review.

In Otsu's method [137] the optimal threshold is selected such that the intra-class variance of background and foreground histograms is minimized. Specifically, if we assume, that the pixel values are integer numbers from interval $[0, n]$ and denote the g-th bin of the histogram as $h(g)$ and variances of background and foreground histograms as $\sigma_B(\lambda_i)$ and $\sigma_F(\lambda_i)$ respectively, the optimal threshold is

$$\begin{aligned} \lambda_i^* &= \operatorname*{argmin}_{\lambda_i} H_B(\lambda_i)\,\sigma_B^2(\lambda_i) + H_F(\lambda_i)\,\sigma_F^2(\lambda_i) \\ H_B(\lambda_i) &= \sum_{g=0}^{\lambda_i - 1} h(g) \\ H_F(\lambda_i) &= \sum_{g=\lambda_i}^{n} h(g). \end{aligned} \tag{2.17}$$

In maximum entropy method [95] the optimal threshold is selected such that the entropy of the background and the foreground histograms is maximized:

$$\lambda_i^* = \underset{\lambda_i}{\operatorname{argmax}} \left(- \sum_{g=0}^{\lambda_i - 1} p_B(g) \log p_B(g) - \sum_{g=\lambda_i}^{n} p_B(g) \log p_B(g) \right) \tag{2.18}$$

$$p_B(g) = \frac{h(g)}{H_B(\lambda_i)}$$

$$p_F(g) = \frac{h(g)}{H_F(\lambda_i)}.$$

Another widely used approach for threshold selection utilizes the k-means algorithm [118]. The histogram bins are interpreted as datapoints and partitioned by the algorithm into two clusters. Segmentation accuracy might be improved by partitioning the histogram into multiple clusters, such that the cluster with the lowest mean is considered as background and the other form the foreground [47].

Since global thresholding is very sensitive to image quality, practical methods often rely heavily on preprocessing. The basic preprocessing may include noise reduction by a Gaussian or median filter, enhancing the contrast, intensity clipping, background subtraction, etc. [196] In [8] a low pass filter is first applied to the image in order to reduce the noise and the resulting image is then re-scaled and subtracted from the raw image. In [212] the preprocessing is derived from a mathematical model of phase contrast microscopy and allows to remove optical artifacts (halos around the cells) inherent to this imaging technique.

The global thresholding methods are fast, easy to implement and many of them are inherently non-parametric, but for many types of data they are too simplistic to provide accurate segmentation. Up to some extent this can be tackled by domain-specific preprocessing, which however compromises the aforementioned advantages. Furthermore some imaging techniques (e.g. fluorescent microscopy) tend to produce images with unimodal histograms, which are unsuitable for global thresholding methods [68].

Adaptive Thresholding

Some of the shortcomings of the global methods can be overcome by choosing a threshold for each pixel separately. A simple option is to split the image into non-overlapping windows and apply a global thresholding method for each window separately [204]. Although this method may outperform global methods e.g. when the illumination changes slowly across the image, the resulting segmentation is likely to suffer from rasterization artifacts.

More advanced methods use sliding window approach and calculate the threshold for each pixel i separately based on properties of the image in the surrounding window. The method of White [202] relates the threshold to the average brightness of pixels in the window. Formally, if we denote the mean of pixel values in the window as $\mu(W_i)$, the threshold is given as

$$\lambda_i^* = \lambda_{Whi} \mu(W_i), \tag{2.19}$$

where λ_{Whi} is a parameter of the method. This approach can be efficiently implemented using the integral image [26].

The method of Niblack [133] uses mean and standard deviation of pixel values in the window:

$$\lambda_i^* = \mu(W_i) + \lambda_{Nib}\sigma(W_i), \tag{2.20}$$

where $\sigma(W_i)$ is the standard deviation of pixel values in the window surrounding pixel i λ_{Nib} is a parameter of the method.

The method of Palumbo [138] can be seen as a combination of global and local approaches. First, an initial segmentation is produced using a global threshold. The method then iterates over (preliminary) foreground pixels and reconsiders their labels as follows. For each pixel the surrounding window is divided into nine subwindows (grid 3×3) and the average brightness of the central subwindow $\mu(W_{kcenter})$ is compared with the average brightness of pixels in the neighboring subwindows $\mu(W_{kneigh})$ (the method uses only the diagonal subwindows and ignores the others). A foreground pixel i is relabeled to background, if

$$\mu(W_{kcenter}) < \lambda_{Pal}\mu(W_{kneigh}) \tag{2.21}$$

where λ_{Pal} is a parameter of the method.

Although adaptive thresholding methods are applicable to wider range of data than the global approaches, their better performance comes at a cost. They tend to be more computationally expensive and often require hyperparameter tuning to work properly. Furthermore even adaptive methods fail to provide reliable detection for certain types of data, e.g. confluent cell cultures.

2.2.2 Region Growing

Region growing methods partition the image into several disjoint regions (each representing one object) by iteratively adding pixels to the initial seeds [1, 219]. The seeds are small regions (in extreme case single pixels) and must be selected carefully, because they determine the number of objects in the image and their rough positions. In every iteration a region growing method update the segmentation as follows. Let us denote the current set of regions as $S = \{S^j \mid j = 1..n\}$ and we use symbol $\mathcal{N}(S^j)$ to denote the set of pixels adjacent to S^j that do not belong to any other region. Every pixel $i \in \mathcal{N}(S^j)$ is assigned a discrepancy score M_i, which quantifies, how different is i from its adjacent region. The method then selects a pixel with smallest discrepancy score adds it to its adjacent region. Pixels that neighbor two or more regions are either labeled as boundary pixels, which do not belong to any region, or their discrepancy score is calculated with respect to the most similar region. The growing process stops either when all pixels were assigned to some region (or labeled as boundary) or no pixel meets a maximum discrepancy threshold defined by the user.

The initial seeds can be selected either manually or automatically. The manual approach is utilized by semiautomatic segmentation methods, which allow the user to interactively refine the seed positions until the segmentation result is satisfactory [91, 98, 148]. Although these methods require only minimal user input, an automatic selection method is needed to make the results repeatable.

The approaches for automatic seed selection can be categorized into several groups [130]. Some methods calculate a feature vector for every pixel in the image and use it to estimate the probability that it belongs to an object of interest. Pixels that meet some predefined criteria (e.g. local maxima of probability) are chosen as the initial seeds [124, 182]. Other approaches utilize edge detection to identify seeds in homogeneous regions with small gray-value gradients [87]. Another option is to first obtain a preliminary segmentation using a different segmentation method (e.g. thresholding) and use the foreground regions as the initial seeds [4, 175].

The exact definition of the discrepancy score M_i is dictated by properties of the typical input images. A straightforward option is to measure difference between the gray level I_i of pixel i and average gray-level of the adjacent region [1]:

$$M_i = |I_i - \mu(I_{S^j})| \,. \tag{2.22}$$

This definition is not convenient for objects that do not appear as homogeneous regions and therefore other approaches define the discrepancy in terms of Gabor filters and other texture features [203].

Watershed Algorithm

Watershed algorithm [21, 22] is arguably the most widely used region growing method for biomedical image segmentation [196, 206]. The intuition behind the algorithm comes from geography. The image is seen as topographic surface, such that the gray-level of each pixel is proportional to its altitude, and the seeds are water sources, that pour water into the landscape. As the water rises, it first floods pixels and valleys (catchment basins) close (in terms of distance and altitude) to the seeds and eventually it floods pixels in higher and higher altitude. If the water pouring from one seed meets with water from another seed, the corresponding pixel is labeled as boundary. Formally, watershed algorithm is a region growing method such that the discrepancy score M_i is defined as the gray-level of the corresponding pixel:

$$M_i = I_i \tag{2.23}$$

(note that I_i might be replaced by a score calculated from a feature vector of pixel i).

The watershed algorithm can be either used directly to obtain the segmentation [210] or as a preprocessing step, which splits the image into superpixels [159]. Due to its simple definition of the discrepancy score (2.23) it can be implemented in linear time with respect to the number of pixels [41].

Region growing methods are suitable for detection and segmentation of homogeneous objects, often allow for efficient implementation and, unlike thresholding methods, are able to separate touching objects. The measure of homogeneity can be defined freely in terms the of discrepancy score between a pixel and an adjacent region, which allows to apply region growing methods even to data, where objects do not appear homogeneous in gray-level representation. However, discrepancy measure developed for certain type of data is in general not suitable for different types, which (along with the seed selection problem) makes a development of a region growing method applicable to wide range of data a tedious task. Region growing methods also tend to be sensitive to noise and require extensive preprocessing.

2.2.3 Deformable Models

Deformable models (also called snakes or active contours) represent the shape of objects by directly modeling their boundary curves – contours (surfaces in 3D) [145]. The theory of deformable models is inspired by physics of a thin flexible membrane, which is subject to various forces and evolves its shape until it reaches an equilibrium. The forces are of two types. The external forces are derived from the image and push the model towards the object boundary. The internal forces serve as a shape prior and keep the boundary smooth. There are two major families of deformable models. The parametric models are represented explicitly as a curve (surface) in the parametric form. Geometric models, on the other hand, are represented as a zero level set of certain higher-dimensional function. Although it can be shown, that these two formulations are to a large extent equivalent [208], they are implemented differently by practical methods, which makes them convenient for different applications. In the rest of this subsection we provide a brief introduction to the underlying theory of deformable models, describe their strengths and weaknesses and review their applications in biomedical image processing.

Parametric Deformable Models

In [99] the parametric deformable model is defined in terms of an energy functional, which reaches the minimum energy when the contour $\mathbf{v}(s) = (x(s), y(s))$ is smooth and resides on edges of object of interest:

$$E_{sn} = \int_0^1 E_{int}\left(\mathbf{v}(s)\right) + E_{ext}\left(\mathbf{v}(s)\right) ds, \tag{2.24}$$

where E_{int} is the internal energy and E_{ext} is the external energy derived from the image. Since finding global minimum of (2.24) is in general intractable, the optimal contour is found by treating the curve $\mathbf{v}(s)$ as a function of time as well – the curve evolves from an initial shape due to internal and external forces and reaches a local minimum of (2.24), when all forces balance each other.

This formulation allows to use only conservative forces, i.e., forces that can be written as gradients of scalar potential functions. In order to incorporate more general forces it is necessary to formulate the model directly as a dynamical system [207]. The dynamics of the curve $\mathbf{v}(s, t)$ can be described by the following equation:

$$\gamma \frac{\partial \mathbf{v}}{\partial t} = \mathbf{F}_{int}(\mathbf{v}) + \mathbf{F}_{ext}(\mathbf{v}), \tag{2.25}$$

where γ is an arbitrary non-negative constant.

The behavior of a deformable model depends on the characteristics of the internal and external forces. In [99] the internal force is defined such that it discourages stretching and bending:

$$\mathbf{F}_{int}(\mathbf{v}) = \alpha \frac{\partial^2 \mathbf{v}}{\partial^2 s} - \beta \frac{\partial^4 \mathbf{v}}{\partial^4 s}, \tag{2.26}$$

where weights α and β control, how much the model resists stretching and bending, respectively. The external force is defined such that the contour is attracted to to nearby edges:

$$\mathbf{F}_{ext}(\mathbf{v}) = -\nabla \left| \nabla (G_\sigma * I)(\mathbf{v}) \right|^2, \tag{2.27}$$

where ∇ denotes the gradient, G_σ is a Gaussian kernel with standard deviation σ and I is the image. The value of σ controls, whether the model is attracted to distant edges (large σ) or it is able to track the object boundary accurately.

To achieve both long range and accuracy, Terzopoulos et al. [187] proposed a multiresolution scheme, which starts with large σ to allow the model to locate coarse boundaries and then iteratively decreases σ in order to capture fine details. Cohen [38] proposed an external force (balloon force), which inflates/deflates the model regardless on the appearance of the image and, in this way, overcomes the problem of tuning initial σ. Xu and Prince [207] proposed an external force called gradient vector flow, which allows the model to locate boundaries of objects with narrow concavities. In order to use deformable models in semiautomatic segmentation methods Kass et al. [99] proposed so called interactive forces that attract the model towards user defined points (spring force) or, conversely, push the model away from points provided by the user (volcano force).

Parametric deformable models enjoy rigorous formulation, but in the same time they have several inherent limitations. Since they parameterize the contour as a closed curve, they cannot be used directly for detection of objects with holes or objects composed of multiple parts. Furthermore, their efficient implementation is complicated due to issues related to discretization of equation (2.25). The forces acting upon the contour may cause it to intersect with itself and fail to extract the correct boundary. As the contour evolves from its initial shape, it must be repeatedly reparameterized in order to maintain the approximation accuracy, which is computationally expensive. The performance of the model may also suffer from numerical instabilities due to discrete time and space.

Geometric Deformable Models

In order to overcome main the limitations of the parametric models Caselles et al. [31] and independently Malladi et al. [121] proposed to represent the contour implicitly as a zero level set of some higher-dimensional function ϕ. A common choice is to define ϕ in terms of a signed distance transform, which determines for each pixel of the image its signed distance from the nearest point of the contour. This distance transform can be calculated efficiently using the fast marching method [173] or the fast sweeping method [217].

The time evolution of ϕ is governed by the following partial differential equation:

$$\frac{\partial \phi}{\partial t} = F|\nabla \phi|, \tag{2.28}$$

where F is a speed function that controls characteristics of time evolution of ϕ and it can depend on various arguments, including the curvature, normal direction, pixel intensities etc. Malladi et al. [121] proposed to use

$$F = \frac{\kappa + F_0}{1 + |\nabla(G_\sigma * I)|}, \tag{2.29}$$

where κ denotes the curvature of the contour and F_0 is a constant. The curvature term in 2.29 serves as a shape prior, which keeps the boundary smooth, and F_0 is equivalent to the balloon force since it causes the contour to inflate ($F_0 < 0$) or shrink ($F_0 > 0$) and the denominator couples the contour evolution with the image data.

Deformable models can be extended in numerous ways. Zhang et al. [216] used Principal Component Analysis to create a hierarchical shape prior, which allows the deformable model to exploit information about global shape of the segmented objects. El-Baz and Gimel'farb [53] combined deformable models with Markov Random Fields, which served as a shape and appearance prior. Their approach allows to learn the characteristics of the speed function from the training data. Martin et al. [123] used probabilistic anatomical atlas to provide reliable initialization for a deformable model, which is then used to find the final segmentation. Bogovic et al. [23] extends the conventional geometric formulation such that it can segment arbitrary number of objects using constant number of level set functions.

Deformable models (both parametric and geometric) are immensely popular in biomedical image processing [57, 126, 193]. They have been used for segmentation of brain tissue [49], lungs [111], livers [115], tumors [10, 164] and other types of data [86, 117] and applied to various modalities, e.g. magnetic resonance [120, 170] or ultrasound images [67, 171, 206]. They owe their popularity to many theoretical and practical advantages, e.g. rigorous formulation, ability to model local properties of the shape (e.g. smoothness) or low computational complexity [127]. On the other hand, since most of the methods rely on gradient descent to find the optimal contour, it is necessary to initialize the model close to the object boundaries to avoid suboptimal local optima. Geometric models may also leak through boundary gaps, which are common in low contrast images (parametric models are more resistant to this issue). Traditional deformable models are also unsuitable for segmentation of objects with sharp tips.

2.2.4 Graphical Models

The first widely used MRF for image segmentation was the Potts model [65]. It is composed of discrete variables, which correspond to individual pixels. Every variable is connected by edges to the neighboring variables such that the resulting graph forms a lattice (Figure 2.1b). Traditionally, the Potts model is used as a segmentation prior in a joint model of the image and the segmentation:

$$p_\theta(x)p_\theta(I \mid x) = \frac{1}{Z(\theta)} \exp\left(\sum_{ij\in E} \psi_{\boldsymbol{\theta}_{ij}}(x_{ij})\right) \prod_{i\in V} p_\theta(I_i \mid x_i). \tag{2.30}$$

The potential functions $\psi_{\boldsymbol{\theta}_{ij}}$ are often translation invariant and defined such that the model penalizes different labels in neighboring pixels:

$$\psi_{\boldsymbol{\theta}_{ij}}(x_{ij}) = \begin{cases} -\theta_P & x_i \neq x_j \\ 0 & x_i = x_j. \end{cases} \tag{2.31}$$

As a result this model prefers segmentations with short boundaries. A simple choice for the appearance model $p_\theta(I_i \mid x_i)$ is a mixture of Gaussians.

Potts model with submodular pairwise potentials is a cornerstone of lots of segmentation methods for biomedical images [5, 11, 34, 33, 142]. This basic approach can be extended in various ways. Roullier et al. [162] uses several Potts models in multiresolution framework (one model per resolution level). Every model is responsible for certain part of the whole segmentation pipeline starting with separation of tissue from slide background in the lowest resolution and ending with segmentation of tumorous cells. Bensch and Ronneberger [17] use image-dependent pairwise potentials, which penalize boundaries in homogeneous regions and makes them more probable in regions with big changes of pixel intensities. The resulting potentials are still submodular and therefore the inference task remains tractable. Wu et al. [205] employs a multilayer CRF with non-submodular potentials for segmentation of cell nuclei. The resulting model is more powerful than the Potts model, which however comes at a cost since the inference task can be solved only approximately. Karimaghaloo et al. [96] uses two CRFs for detection and segmentation of small enhanced pathology in medical images – a pixel-level segmentation model with ternary potentials and a patch-level CRF with latent variables, which serve as shape priors.

Standard Markov Random Fields are not convenient for detection of touching objects (e.g. cells in confluent culture), because they can only provide semantic segmentation. Memariani et al. [131] tackle this problem by using two CRFs, one for background/foreground segmentation and another one for segmentation of boundary and non-boundary pixels. Pan et al. [139] use a multilabel CRF for detection of cell centroids such that every variable has in general different labelset. The labelsets are created dynamically and contain information about the cell id, which allows to separate touching foreground regions.

Markov Random Fields as Shape Priors

Standard pixel-level Potts model can be seen as a very simple shape prior, which prefers objects with short boundaries. However, it cannot model more complicated shape assumptions (e.g. locally smooth boundaries, sharp tips) or even represent a global shape prior (this is useful for objects with characteristic shape, e.g. lungs in frontal X-ray radiographs or certain types of cells). This can be only achieved by models with higher order potentials. Such models are however impractical, because the number of parameters of potential functions grows exponentially with number of variables in the corresponding cliques, which makes both inference and parameter learning quickly intractable.

One option how to circumvent this problem is to represent the potentials using smaller number of parameters. This approach is used by the Fields of experts model proposed by Roth and Black [160], which represents the potentials as (non-linear) combinations of responses of linear filters. The higher order potentials can be also represented by latent variables. A simple, yet effective example of models of this type are MRFs on bipartite graphs, such that one layer is composed entirely of visible variables and the other layer contains latent variables only (Figure 2.1c). The bipartite models can be used as global or local shape priors depending on connectivity of the latent variables. Fully connected models (also known as Restricted Boltzmann Machines [79, 92]) are suitable for global priors and partially connected models can be used as priors for fixed regions of the image [55]. Local shape properties can be modeled by translation invariant models, such that every latent variable is connected to visible variables in its receptive field and parameters of corresponding edges are shared across the model [2]. A bipartite model can contain latent variables of all types and consequently serve as both global and local prior. Bipartite models can be further generalized to n-partite graphs. These models are called Deep Boltzmann Machines [35, 167].

2.2.5 Patch Classification

Patch-based methods utilize sliding window approach and classify each pixel using only a local patch around that pixel. Unlike some previously discussed approaches, patch-based methods do not enjoy rigorous mathematical background, but they are easy to implement and provide decent performance on various types of data [13, 74].

The patch-based methods are composed of two main steps: feature extraction and patch classification. Different methods can be then distinguished by types of features they use to characterize the patches and the classifier utilize for their classification. Kainz et al. [93] used random forests for detection of cell centroids. The feature vectors included features derived from RGB and

Luv pixel intensities, gradient-based and Haar-like features. Random forests were also employed by Geremia et al. [66] for segmentation of brain tumors in multi-modal magnetic resonance images. Dahl and Larsen [42] proposed to learn a visual dictionary of patch-label pairs from the training data and use nearest neighbor classifier to obtain semantic segmentation. Tong et al. [191] designed features based on patch similarities and used along with Support Vector Machines (SVM) in order to estimate progression of Alzheimer's disease from structural magnetic resonance images. Huh and Chen [84] used SVM and Hierarchical Conditional Random Fields to detect mitotic events in sequences of phase-contrast microscopy images. Since mitotic events typically happen across several frames, the patches cover 3D volume, which spans over several frames. Hou et al. [83] used convolutional neural network for patch-based semantic segmentation directly from the image data. A similar approach was adopted by Varghese et al. [6] with generative adversarial networks.

The main advantages of patch-based methods are their performance in semantic segmentation, straightforward implementation and modularity, i.e., they allow to change the types of features of the classifier without affecting the other part of the method. However, similar to thresholding methods, they do not allow easy integration of a shape prior and other methods must be used to e.g. enforce short boundaries [12]. Furthermore standard patch-based method do not have a natural mechanism for splitting touching objects of the same class, which must be delegated to the postprocessing [104].

2.2.6 Neural Networks

Artificial neural networks are mathematical models loosely inspired by biological neural networks. They are composed of large number of interconnected computational units (neurons) typically organized into layers. The connections represent the flow of information in the network – the neurons receive their inputs via connections to the previous layer(s) and connections to the following layer(s) distribute their outputs to other neurons. Neural networks where information propagate in one direction only are called feed-forward, whereas networks that contain at least one loop are called recurrent.

The neurons take linear combination of their inputs x_i and transform them using a non-linear activation function f into a single output:

$$y = f\left(\sum_i \theta_i x_i + \theta_b\right), \tag{2.32}$$

where θ_i is a weight associated with the i-th input and θ_b is called bias and serves as a constant input into the neuron. There are many types of possible activation functions. Particularly popular are sigmoid and rectified linear unit (ReLU):

$$f_{sigmoid}(z) = \frac{1}{1+e^{-z}} \tag{2.33}$$

$$f_{ReLU}(z) = \max(0, z). \tag{2.34}$$

It can be shown that under certain mild conditions a feed-forward neural network with one hidden layer and sufficient number of neurons can approximate any continuous function [82]. The required number can be however very large and it can be shown, that for certain classes of functions deep (multilayer) networks with moderate number of neurons can achieve the same approximation error as shallow networks with exponentially more neurons [54, 112, 132]. It has been also observed empirically that deep networks perform better on many tasks of artificial intelligence than their shallow counterparts [136, 179, 213].

There are many types of neural networks. On of the simplest but nevertheless powerful models are multilayer perceptrons (MLP) [165]. The MLPs are composed of a single input layer, two or more hidden layers (hence multilayer) and one output layer. The neurons in hidden layers are connected with all neurons in the previous and the next layer and therefore their architecture can be represented by a fully connected n-partite graph. The MLPs can be used e.g. for image recognition, which makes them suitable for patch classification methods.

Arguably the most influential type of neural networks used in computer vision are convolutional neural networks (CNN) [77, 105, 109, 157, 179]. CNNs are typically composed of several types of layers, including convolutional, pooling (subsampling) and fully connected layers. The neurons in a convolutional layer are arranged into a $(d+1)$-dimensional volume in the same way as voxels in an d-dimensional image – there are d spatial dimensions and the last dimension is the number of channels. Every neuron is connected only to a small part of the previous layer, which belongs to its receptive field. Furthermore they share their parameters such that all neurons from the same channel have identical parameter vectors, i.e., the layer is translation invariant. Thus, every channel of a convolutional layer can be seen as a convolution of the input data with a filter represented by the input weights of the neurons.

Due to parameter sharing and translation invariance, CNNs for semantic segmentation can be trained end-to-end (i.e., the segmentation is estimated by the network alone) and applied to images of arbitrary size. The neurons in a trained CNN represent certain features of the image. For example, the neurons in the first layer may activate in presence of various oriented edges or color blobs, whereas the neurons in the subsequent layers may represent higher order features like object parts and their topology. The convolutions can be also implemented more efficiently than a naive sliding window approach, which makes CNNs in general superior to the patch classification methods.

Automatic parameter learning is a key element of neural networks, because even a relatively small network may have millions of parameters. The task of supervised parameter learning can be formulated as minimization of a loss function, which measures discrepancy between the actual and expected (ground truth) output of the network, and can be solved by a gradient descent algorithm [101]. The gradient is calculated using methods of automatic differentiation, which represent the network as an oriented graph where every node corresponds to a single computation (e.g. summation, multiplication, exp, etc.) and repeatedly apply the chain rule to obtain the gradient [15]. This allows to train networks with arbitrary structure without need to derive the learning task manually.

Due to the large number of parameters the neural networks are prone to overfitting and there is a number of techniques which help the network to generalize. Some of them are applicable even to other types of models, for example weight regularization, which penalizes large weights in favor of smaller ones. Another technique specific to neural networks is Dropout [183]: In every training epoch a random subset of neurons is (temporarily) removed from the network and their weights are not updated. This technique is inspired by ensemble learning, which aims at training several different models in parallel and at the end combine them into one final model [73]. The risk of overfitting can be also decreased by increasing the amount of the training data. Unless more data are acquired, this can be achieved by data augmentation, i.e., by random geometrical and intensity transforms [144]. Data augmentation also helps the network to become invariant to the used transforms. This technique is particularly important in the context of biomedical image processing, because obtaining ground truth annotations is labor-intensive and time-consuming [157].

Neural networks are widely used in computer vision only for the last decade [80] but since then they achieved state-of-the-art performance on various public benchmarks [74, 166, 196]. Traditionally, they were used as classifiers, which take an image as an input and estimate its class (e.g. carcinoma and non-carcinoma) [7, 56]. However, they can also be used for pixel-level segmentation and trained end-to-end, which was empirically shown to be superior to patch classification [157]. There are various neural network based methods for semantic segmentation, which differ mainly by the architecture of the network (number of neurons and their connectivity) and techniques used for parameter learning [75, 157]. Some works utilize already pretrained networks and tailor them for their data by changing and retraining only a few final layers [30].

Recently, there have been numerous attempts to design a neural network for multiple object detection. Faster R-CNN model [154] is composed of two modules. The first module proposes detection candidates and the second module is a classifier of the proposed candidates. The detection candidates are represented by their bounding boxes and their coordinates are predicted directly by neurons in the output layer. Mask R-CNN model [76] extends Faster R-CNN by another module, which predicts for each detection candidate its segmentation mask. The YOLO architecture [152] integrates the proposal and classification modules into one network, which detects the objects (the bounding boxes) by an order of magnitude faster than Faster R-CNN.

Neural networks are the key element of many state-of-the-art computer vision methods. They have several conceptual advantages over other methods discussed in this section – they do not require hand-crafted features, allow for tractable inference and parameter learning and can be used for segmentation and detection of arbitrary objects (as long as there is enough training data). However, nowadays there is no widely accepted approach for selecting the structure of the network and consequently development of new architectures relies on personal experience of the researcher and trial-and-error approach. Neural networks are also memory intensive and require specialized hardware (GPU) for decent performance. Recent research however gives hope, that in the near future it would be possible to use even very deep models on low-end devices [36, 37, 46].

2.3 Tracking

Methods for multiple object tracking can be broadly classified into two categories: tracking-by-model-evolution and tracking-by-detection [110, 116]. In the model-evolution paradigm the objects are detected in the first frame and then tracked throughout the sequence by updating their corresponding models. In the tracking-by-detection strategy the objects are first detected in all frames and then linked together into the final lineage. Although this thesis deals primarily with the tracking-by-detection methods, in this section we provide an overview of both paradigms in order to compare their strengths and weaknesses.

2.3.1 Tracking-by-model-evolution

Tracking-by-model-evolution methods detect the objects of interest in the first frame and represent their positions and shapes by a mathematical model, which is then propagated until the end of the sequence. In each frame the position and the shape of the model is updated to match the movements of the object, taking the model from the previous frame as an initial estimate.

Methods developed for biomedical images represent the objects almost exclusively by deformable models. As discussed in subsection 2.2.3, deformable models were found suitable for detection and segmentation of many types of biomedical objects, but at the same time both parametric and geometric models have their own weaknesses. Parametric models do not address the problem of initial detection, but instead assume, that a rough segmentation of objects of interest is already available. Traditional parametric models are also not well suited for handling splitting objects (e.g. cells undergoing mitosis). Geometric objects, on the other hand, can handle splitting objects naturally and can be also used for the initial detection step, because they allow to model multiple objects by one level set function. However, when two objects touch, their associated fronts merge and they are mistakenly considered as one object, which makes geometric models unsuitable for high density data (e.g. confluent cell cultures). Practical methods must also address some tracking-related issues: recovery of lost tracks, detection of objects that do not appear in the first frame and incorporation of a motion model.

Dufour et al. [50] use for every tracked object its own level set function and constrain their evolution by an overlap penalty. The objects in the first frame are detected by a separate geometric deformable model using a convention that every zero level set delineates a contour of one object. The same approach is used to detect objects, that move into the field of view after the first frame. Zimmer et al. [218] employ parametric snakes (one snake for each object) and handle object splitting by topological operators, which split the model into two when it develops a bottleneck geometry. Dzyubachyk et al. [51] tackle the initial detection step by a single level set approach similar to [50] and in addition use a combination of watershed transform and region merging to separate touching objects. Ray et al. [151] couple parametric snakes with Kalman filter, which allows them to use more complex motion model then simply using position and shape of objects as an estimate of their states in the next frame. They also enrich the snake energy by additional terms that regularize the shape of the snake based on both prior knowledge and shape of the object in the previous frames.

The idea of tracking-by-model-evolution methods is based on the assumption, that propagating and evolving a shape of an object throughout a sequence of images is more efficient then detecting the object in each frame separately. Since the objects are detected only in the first frame, the tracker remains fast even if the detection step is computationally expensive. However, the relevance of this argument diminished greatly with recent development of fast and robust object detectors and as a result state-of-the-art methods for multiple object tracking are based mostly on the tracking-by-detection paradigm [196].

2.3.2 Tracking-by-detection

Tracking-by-detection methods (sometimes also called tracking-by-segmentation or tracking-by-assignment) consist of two steps. In the first step a detection module tries to locate positions and shapes of individual objects and generate detection candidates. In the second step (data association) the candidates are linked together into the final lineage.

Tracking methods can be further divided into two groups depending whether they track the objects on frame by frame basis (online) or use the whole sequence at once (offline). The online methods (also called sequential) are suitable for real-time tracking, because they construct the lineage on the fly using only information available up to the current frame. Offline methods can be in general more accurate, because they can use information from the future frames, but in the same time they tend to be more computationally intensive. In the context of biomedical image processing the accuracy of results is often more important than the speed of the tracker, because the objects of interest move slowly and one time step in the video may correspond to several minutes in real time [29]. Division to online/offline approaches also applies to methods based on model evolution, but since most of them are naturally sequential, it is more relevant for tracking-by-detection paradigm.

The performance of tracking-by-detection methods depends heavily on the quality of the detector, because detection errors propagate to the linking step. This can be partially mitigated by generating an overcomplete set of detection candidates and letting the data association module select the optimal subset. This solution however does not deal with other sources of errors, e.g. low signal-to-noise ratio or poor visibility of object boundaries in densely populated areas.

The data association module estimates trajectories of individual objects by linking the detection candidates together. The linking process is primarily based on their appearance, but in addition it is often guided by various submodels, which estimate positions and shapes of objects using their states in the neighboring frames. A typical example is a motion model, which models dynamic behavior of individual objects. Dynamics of biological objects can be often modeled using simple assumptions, e.g. Brownian motion, constant velocity or their combination [135]. A motion model is essential in scenarios where objects of interest cannot be distinguished by their appearance. Some tracking methods also employ a set of constraints in order to forbid physically impossible events. For example it is not possible for a pair of cells in a monolayer culture to switch their positions by "jumping" over each other and for the same reason one detection candidate cannot be shared by multiple trajectories. Other examples of data association submodels are interaction models, which model the tendency of certain objects to form groups, to keep fixed distance or to repulse each other. Interaction models were applied to pedestrian tacking [143, 149, 209] and are also relevant for many types of biomedical data (e.g. migrating cells).

The data association task is typically formulated as an optimization problem over possible lineages, such that the best lineage has minimal cost. The cost associated with a lineage specifies how well the linked detection candidates fit together. The cost of linking a detection candidate with an existing trajectory can be defined in many different ways, which can be broadly classified into three categories: appearance based, motion based and combination of both.

The appearance based approaches represent the candidate by various features, which are then compared with the features representing the object in the neighboring frame(s) [116]. The features might be based on color (e.g. average intensity of pixels, color histogram), gradients (e.g. SIFT [114], histogram of oriented gradients [44]) or texture (e.g. region covariance matrix [194], local binary patterns [134]). The visual features are widely used by general purpose tracking methods, but their value for biomedical images is limited, because due to low microscope resolution or used modality the objects of interest are often inherently visually indistinguishable.

Motion based approaches define the cost as the discrepancy between the position and shape of the detection candidate and the object state predicted by the motion model. For example if the state of the object is specified by a vector of numerical values (e.g. coordinates of the centroid, radius, etc.), the discrepancy measure can be defined as a weighted sum of differences of individual parameters [94]

$$\sum_i \lambda_i \|\theta_i - \hat{\theta}_i\|, \tag{2.35}$$

where θ_i are the parameters of the detection candidate, $\hat{\theta}_i$ are the parameters predicted by the motion model and λ_i are the weights.

Unconstrained data association is a difficult combinatorial problem. Even if considered for two frames only, there are $\mathcal{O}(n!)$ possibilities to pair the detection candidates. In some types of data (e.g. proliferating cells) the detection candidates in the first frame might be even linked with two candidates in the second frame in order to represent cell division, which further adds to the complexity of the problem. There are however some special cases, where globally optimal linking can be found efficiently: if every detection candidate corresponds to exactly one object and their number is constant, the assignment problem can be solved by the Hungarian algorithm in polynomial time [106]. The first step is to calculate a cost matrix such that each entry represents a penalty of linking the corresponding pair of detection candidates (one from the first frame and the other from the second frame) into a trajectory. The algorithm is based on observation, that subtracting a number from all entries of one row or column does not change the optimal assignment. Initially, it subtracts the smallest entry in each row from all the entries in that row, which is then followed by subtracting smallest entries for each column. After this operation several entries in the cost matrix will become zero. The next step is to find a minimal subset of rows and columns containing all the zeros in the matrix. If the size of the subset equals to n, the algorithm ends and the optimal assignment is given by the zero entries of the matrix. If the number is smaller, the algorithm finds the smallest element that does not belong to any selected row or column, subtracts it from all entries in non-selected rows and adds it to all entries of selected columns. The selection and subtraction steps are repeated until all zeros in the matrix can be covered by selecting exactly n rows and/or columns. The time complexity of this algorithm is $\mathcal{O}(n^4)$ but it can be modified to achieve $\mathcal{O}(n^3)$ running time [52].

One of the simplest, yet widely used approaches for general data association is a greedy algorithm. It links the detection candidates in nearest-neighbor manner, starting from the first frame and gradually progressing to the end of the video sequence, which makes it inherently sequential even if the whole sequence is available beforehand. There are many different variants that share the same basic principle. Goulian and Simon [71] used a no-choice greedy algorithm, which in each frame prolongs trajectories of existing objects using only detection candidates within a fixed sized window centered in the last position of the objects. The trajectory is prolonged only if the window contains a single detection candidate. If there is no candidate, or the window contains more than one, the trajectory is broken. This algorithm is simple to implement, but it is only suitable for low density data.

The most widely used variant of the greedy algorithm can be described as follows. Let us assume, that the algorithm already linked trajectories up to frame $t-1$ and proceeds to frame t. For every object, whose trajectory ends in frame $t-1$, and every detection candidate in frame t it calculates the cost of appending the candidate to the trajectory. For many types of data it is reasonable to consider only pairs that are closer to each other than some predefined threshold, which represents the maximum velocity of objects of interest. The pairs are sorted according to the cost and the trajectories are prolonged one after another using detection candidates with smallest possible cost, i.e., the trajectories are prolonged by the nearest candidate (in terms of the cost), which is not already assigned to another trajectory. Alternatively, the trajectories may be prolonged only if the cost of the best available detection candidate is smaller, than a predefined threshold. This is useful for high density data, where the objects may disappear instantly (die, move out of focus) and new objects may emerge by chance in the next frame in their vicinity.

A better total cost can be achieved by the greedy exchange algorithm [172]. After all trajectories are prolonged to the frame t, it iterates over all pairs of trajectories and for each pair tries to exchange their detection candidates in frame t. If the swap decreases the total cost, the trajectories are changed accordingly. The algorithm continues until the total cost cannot be further improved. A better local optimum can be found by generalizing the exchange operation to triples or even more trajectories, which however significantly increases the time complexity of the algorithm.

Instead of constructing lineage in frame-by-frame manner, greedy algorithms can be also applied trajectory-wise, i.e., the final lineage can be constructed by adding trajectories one at a time. This approach was used by Magnusson et al. [119] for tracking cells in bright-field microscopy images. They represent the final lineage as an acyclic graph, such that every node corresponds to certain spatially localized event, e.g. migration of a single cell between two frames, mitosis, apoptosis, etc. Every event is associated with a single frame and therefore linking the corresponding nodes allows to reconstruct the state of the cell culture in every time step of the video sequence. The events are of three types: migration, existential and modification events. The migration events represent a migration of one cell between two frames, either from one location to another, out of the field of view or into the field of view. The existential events represent either mitosis (cell division), or apoptosis (cell death). The modification events are used to modify the lineage created so far. The cell count event allows to indicate, that a detection candidate does not correspond to a single cell only, but to several of them (one more than in the previous lineage). The separation event allows to cut an already existing trajectory and used one of its parts which is currently being constructed. The total cost of a lineage is defined as a sum of contributions of events used to construct the lineage. The final lineage is created by iteratively adding new trajectories with the minimum cost. Adding a new trajectory is equivalent to finding a shortest path in an n-partite graph, which can be efficiently solved by the Viterbi algorithm [198]. The Viterbi algorithm also allows to incorporate various constraints which ensure, that the resulting lineage has a biologically valid interpretation, e.g. a mitosis node can be used by a new trajectory only if there is another trajectory, which represents the parent and the sister cell. The single-event costs are allowed to be negative, which provides a natural way for estimating the optimal number of trajectories – the algorithm stops, when adding a new trajectory cannot decrease the total cost.

Multiple hypothesis tracking (MHT) [153] is an online data association method, which constructs multiple trajectories (hypotheses) for each object and delays the final decision until the linking ambiguities can be resolved. The set of trajectories associated with one object can be seen as a tree, which originates in a single detection candidate. In each frame the trajectories in the tree are expanded with detection candidates that fulfill all data specific constraints (e.g. maximum velocity). Since the number of trajectories in each tree grows exponentially, various pruning techniques are necessary to keep their number manageable. Common approach is to keep only m best hypotheses and in addition restrict number of detection candidates used to expand each trajectory by a threshold [100]. Final decision (finding the best lineage) can be formulated as a maximum weighted independent set problem, which can be solved by an approximate algorithm [28, 140].

Zhang et al.[215] formulate the data association task as a min-cost/max-flow problem. The flow graph representing the lineage is defined as follows. Every detection candidate is represented by two detection variables (incoming and outgoing) connected by an edge. The amount of flow (a nonnegative integer number) passing through this edge indicates the number of objects represented by the corresponding detection candidate. Since their model disallows occlusions, the capacity of the edges was limited to 1. The outgoing detection variables are connected to several incoming detection variables in the next frame. These edges have also capacity 1 and represent possible moves of objects between frames. Incoming variables in all frames are further connected to the source node and the outgoing variables to the sink. The optimal lineage is found by sending k flows from the source to the sink, where k is an a priori unknown number of objects. The flow conservation principle and capacity of edges limited to 1 ensure that the solution consists of node-disjoint paths, which allows to reconstruct the trajectory of each object unambiguously. The optimal flow is found using an off-the-shelf push-relabel method [70]. Berclaz et al. [18] exploited the special structure of the problem to find the optimal solution more efficiently. They reformulated the min-cost/max-flow problem as an integer linear program and proved that its linear program relaxation yields the optimal solution. Furthermore, they demonstrated, that the relaxation can be solved by the k-shortest paths algorithm [185]. Pirsiavash et al. [147] proposed an alternative algorithm, which finds the optimal solution by solving $k+1$ shortest path problems. Turetken et al. [195] generalized the basic min-cut/max-flow formulation for tracking biological cells. They construct the flow network using several types of variables, which represent events characteristic for a cell population, including cell division and chunk separation, which allows to separate confluent cells with poorly visible boundaries.

Many approaches formulate the data association task in a probabilistic framework. Joint probabilistic data association filter (JPDAF) [61] is a data association method for fixed number of objects. Since it is an online method, it deals with a probability distribution of all feasible assignments of detection candidates (measurements) to objects given their state in the previous frames. Instead of choosing the most probable assignment it calculates the conditional marginal probability of states of individual objects and uses its expected value to update their states:

$$\rho^{(t)*} = \mathbb{E}\left[\rho^{(t)} \mid R^{(t)}, M^{(t)}\right] \tag{2.36}$$

where $\rho^{(t)}$ denotes the state of object ρ in frame t, $R^{(t)}$ are states of objects up to frame $t-1$ and $M^{(t)}$ are the detection candidates in frame t. The object states are represented explicitly as numerical vector, which contain information about position, size, velocity, etc. and therefore the expected value is well defined. A naive implementation of JPDAF is tractable only for a small number of objects, because calculation of marginal probabilities requires to sum over all possible assignments. Methods for tracking higher number of objects must therefore rely on approximations. A common option is to use a particle filter, which approximates the marginals by generating weighted samples of possible assignments [88]. Rezatofighi et al. [156] proposed to approximate the marginals using m best assignments and showed, that under certain realistic assumptions these assignments can be found using linear programming. Despite being limited to a fixed number of objects JPDAF has been used e.g. for tracking objects in time-lapse fluorescence microscopy [69, 155].

Probabilistic graphical models were used in various ways for the data association task. Chakraborty and Roy-Chowdhury [32] proposed an online tracking method, which employs Conditional Random Fields (CRF) for linking objects with the detection candidates in the next frame. The CRF is constructed on the fly for each frame separately. Its nodes correspond to the objects and their labels represent indices of detection candidates in the next frame that can be used to prolong their trajectories. A special label is used to encode that the trajectory ends. Neighboring nodes are connected by edges that represent mutual interactions of the objects and ensure, that one detection candidate is linked with at most one object. The optimal labelling (and consequently assignment) is found using loopy belief propagation.

Schiegg et al. [168] proposed a graphical model for offline tracking with two types of binary variables: detection variables, which correspond to the detection candidates, and transition variables, which represent links between the candidates and allow to reconstruct the final lineage. The probability of each lineage is defined using three types of higher order potentials. The detection potentials are associated with every set of conflicting (overlapping) detection candidates and their value is either a score, which quantifies the belief of the detector that the selected candidate corresponds to a single object, or a penalty if none of them is selected. The count potentials are associated with detection candidates that form a connected component and depend on the number of objects within that connected component. The transition potentials incorporate a single detection variable along with all associated transition variables and their value depends on the behavior of the corresponding object, i.e., whether it disappears, migrates or splits. Furthermore the model consists of 0-1 constraints which ensure, that the lineage is feasible. The inference task is formulated as an Integer Linear Program and solved by an off-the-shelf solver.

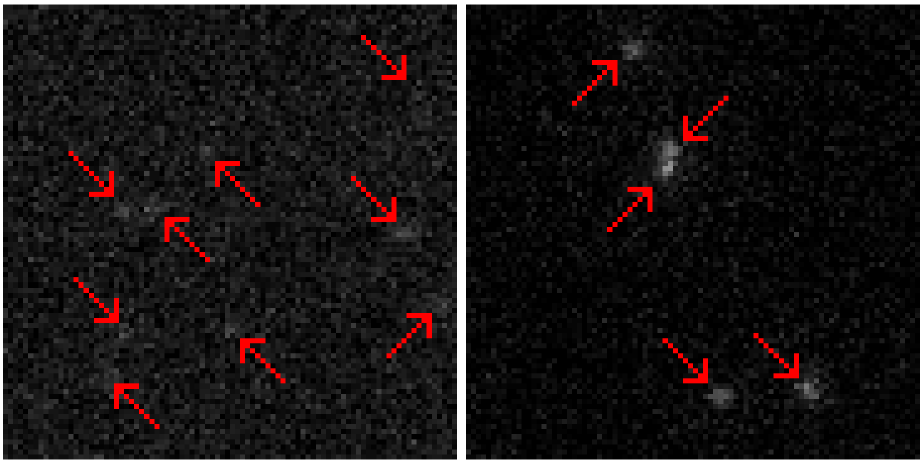
Li et al. [110] combined tracking-by-model-evolution and tracking-by-detection paradigms into one online tracking method. The objects are tracked by propagating a deformable model and furthermore in each frame the detection candidates are generated by a patch classifier detector. The object states are updated by comparing the result of the detector with object models propagated from the previous frame. Detection candidates that do not overlap with any propagated model are considered new objects and the method uses them as initial state of their trajectories. If a propagated model does not overlap with any detection candidate, the trajectory of the corresponding object is terminated. If a propagated model overlaps with exactly one detection candidate, the trajectory of the corresponding object is prolonged using that candidate. And finally, propagated models that overlap with more than one detection candidate are analyzed by the track compiler, which decides, whether the corresponding objects divided into two children or should be linked with one of the candidates.

Chapter 3

Detection and Tracking Dimensionless Objects

In this chapter we present a tracking method, which follows the general scheme outlined in chapter 2 – the tracking task is modeled using a probabilistic graphical model and the final tracking result is obtained by minimizing the Bayes risk. It is a preliminary step towards the main goal of this thesis – the detector is integrated only indirectly and the method is able to track dimensionless objects only. Nevertheless, tracking dimensionless particles is an important task in the context of biomedical image analysis, because in many biological objects appear naturally in microscopy images as small points, either because the size of the objects is near the resolution limits of the microscope (e.g. viruses observed by a fluorescent microscope) or because the shapes and internal structure of the objects are not relevant for evaluation of the experiment (this might be the case e.g. for analysis of intracellular transport).

The challenges of tracking dimensionless particles are illustrated in Figure



(a) : SNR 2

(b) : SNR 4

Figure 3.1: Sample images with dimensionless particles (simulated fluorescence microscopy images of vesicles in the cytoplasm). The images have two different signal-to-noise ratios: 2 and 4. The objects are marked by red arrows.

3.1. The objects appear as brighter blobs on a dark background but besides that they have no characteristic texture or color. This makes their tracking, in a way, more challenging, because the tracker cannot rely on appearance based hints to associate the objects across the frames. A good motion model is especially important when two or more objects get close to each other. In some cases spatially close objects interact with each other – they form a permanent group or repulse each other. Modeling these interactions explicitly is beneficial as well, because it provides additional hints for resolving ambiguities. Data with low signal-to-noise ratio (SNR) pose an additional challenge. Since each object consists of a few pixels only, the noise may sometimes by chance erase an existing object from the image or, conversely, create false objects. These types of errors cannot be in principle resolved by the detector alone but require analysis of several neighboring frames.

To address the aforementioned challenges, we propose a probabilistic graphical model for tracking dimensionless objects, which allows to integrate the appearance, motion and interaction submodels into one joint model. To make it more robust for low SNR data, it does not assume, that a detector has generated a fixed set of detection candidates, but instead uses a probabilistic detection map (pixel-wise detection scores) and determines the number and positions of objects by itself. The performance of the model is evaluated on data from the 2012 ISBI Particle Tracking Challenge [135].

3.1 Probabilistic Model

We assume that we are given a sequence of images $I = (I^{(1)}, I^{(2)}, ..., I^{(n)})$ and measurements $M = \{M^{(1)}, M^{(2)}, ..., M^{(n)}\}$, which specify for each pixel a normalized detection score (a number from interval $[0, 1]$) of presence of an object in that pixel. However, they can not be seen as probabilities – $M_i^{(t)} = 1$ does not necessarily imply the presence of an object at pixel i and $M_i^{(t)} = 0$ does not exclude an object in that position.

The basic building block of the model is a trajectory ρ of a single particle.

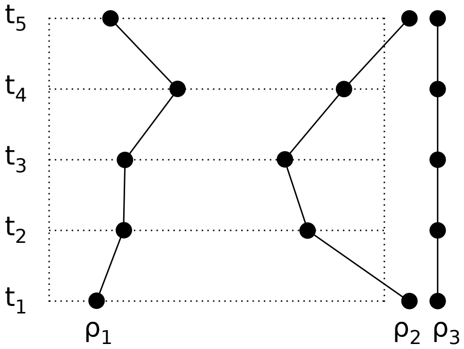


Figure 3.2: Example of object trajectories. The trajectory ρ_1 stays in the field of view in all depicted frames, ρ_2 enters in frame 2 and leaves in frame 4 and ρ_3 is composed solely of dummy positions i_{out}.

It is a sequence of n positions representing the position of the object in corresponding frames. A special "dummy" position i_{out} is used to indicate, that the object is not visible in the scene, because it drifted out of the field of view or does not exist in the corresponding time step (Figure 3.2).

Every variable of the model corresponds to a single trajectory (thus their label is a sequence of n positions). Consequently the number of variables should be higher than the number of objects in the input sequence, because otherwise the model would not be capable to recover trajectories of all objects. Some variables might be left unused in the sense that their label is composed entirely of dummy positions and therefore they do not correspond to any physical object.

To account for the uncertainty inherent to microscopy imaging we propose to model the tracking task by a probabilistic graphical model for sets of trajectories $R = \{\rho_k \mid k = 1..m\}$ given the measurements M. The probability distribution function is defined in terms of potential functions φ_θ for individual trajectories, interaction potentials ψ_θ and hard constraints Γ that ensure, that the set of trajectories is physically possible:

$$p_\theta(R|M) = \frac{\exp\left(\sum_{\rho\in R}\varphi_\theta(\rho, M) + \sum_{\rho,\rho'\in R}\psi_\theta(\rho,\rho') + \sum_{\rho\in R}\Gamma(\rho)\right)}{Z(\theta)}, \quad (3.1)$$

where θ denotes the set of parameters and $Z(\theta)$ is the partition function.

Every unary potential φ_θ is a sum of three potential functions, each related to certain aspect of the trajectory:

$$\varphi_\theta(\rho, M) = \varphi_{\theta det}(\rho, M) + \varphi_{\theta mot}(\rho) + \varphi_{\theta pen}(\rho). \quad (3.2)$$

The detection related potential $\varphi_{\theta det}$ is composed of weighted detection scores for pixels occupied by trajectory ρ:

$$\varphi_{\theta det}(\rho, M) = \sum_{t\in 1..n:\rho^{(t)}\neq i_{out}} \theta_a M^{(t)}\left(\rho^{(t)}\right) + \theta_b, \quad (3.3)$$

where $\rho^{(t)}$ denotes the position of trajectory ρ in frame t and θ_a, θ_b are parameters. Potentials $\varphi_{\theta mot}$ are related to motion and quantify difference between real position of ρ in certain frame and a position predicted by a motion model:

$$\varphi_{\theta mot}(\rho) = -\sum_{t\in 1..n:\rho^{(t)}\neq i_{out}} \left\|\rho^{(t)} - \rho^{(t)}_{m\theta}\right\|^2, \quad (3.4)$$

where $\rho^{(t)}_{m\theta}$ is the position predicted by the motion model. Potentials $\varphi_{\theta pen}(\rho)$ are used to penalize trajectories that enter the field of view in at least one frame: $\varphi_{\theta pen}(\rho) = 0$ if ρ is composed of dummy positions only, otherwise $\varphi_{\theta pen}(\rho) = \theta_c$. This prevents the tracker to model one object by several short trajectories.

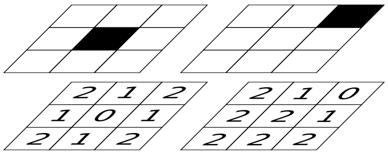


Figure 3.3: Baddeley's loss for two binary images. The binary images are in the first row and their distance transforms ($\lambda_{trunc} = 2$) directly beneath them. The value of the loss is obtained as a sum of squared differences of corresponding pixels of the distance transform. In this example the value is 10.

The interaction (pairwise) potentials $\psi_\theta(\rho, \rho')$ prevent trajectories from getting too close to each other:

$$\psi_\theta(\rho, \rho') = \sum_{t \in 1..n} \begin{cases} 0 & \text{if } \rho^{(t)} = i_{out} \text{ or } {\rho'}^{(t)} = i_{out} \\ -\frac{\theta_d}{\left\| \rho^{(t)} - {\rho'}^{(t)} \right\|} & \left\| \rho^{(t)} - {\rho'}^{(t)} \right\| \leq \lambda_{trunc} \\ 0 & \text{otherwise}, \end{cases} \tag{3.5}$$

where θ_d is a parameter and $\lambda_{trunc} > 0$ is a truncation constant. The motivation for defining the potential in this form is twofold. Since the distance between the objects is in the denominator, the potential penalizes objects closer to each other than θ_d severely but attains much smaller value for objects further apart. In extreme case when two objects overlap completely, the value of the potential is $-\infty$, which makes such configuration impossible. The interaction potentials eliminate the need for rather heuristic non-maximum suppression.

The hard constraints $\Gamma(\rho)$ are $0/-\infty$ valued functions used to ensure, that no object exceeds some predefined maximum velocity.

3.2 The Inference

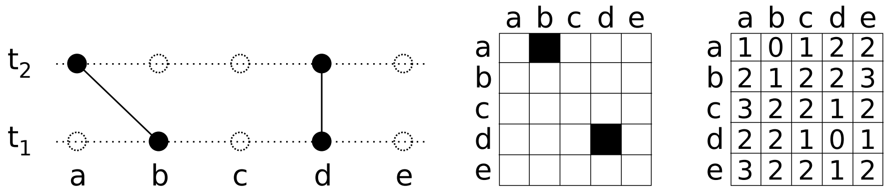
In the context of the Bayes risk minimization a good loss function should reflect the properties of the physical system. From this point of view the commonly used zero-one loss is not a proper choice, because it cannot distinguish between solutions that are only slightly off and completely wrong. Another common choice, the additive L_2 loss, is also inappropriate, because it is unclear how to evaluate the loss for two sets of trajectories with different cardinality.

The loss function we propose is based on the Baddeley's Delta Loss, which was originally developed for binary images [163]. Let B be a binary image and $DT(B)$ its distance transform with respect to a truncated pixel distance

$$dist(i, i'; \lambda_{trunc}) = \min\{\left\| i - i' \right\|_1, \lambda_{trunc}\}, \tag{3.6}$$

where $\|.\|_1$ denotes the L^1 norm. The value of the Baddeley's loss for two binary images B and $\hat{B}$ is then equivalent to the value of additive L_2 loss of their distance transforms (Figure 3.3):

$$l_{Badd}(B, \hat{B}) = l_2(DT(B), DT(\hat{B})) = \sum_i \left(DT(B)_i - DT(\hat{B})_i \right)^2, \tag{3.7}$$



	a	b	c	d	e
a		■			
b					
c					
d				■	
e					

	a	b	c	d	e
a	1	0	1	2	2
b	2	1	2	2	3
c	3	2	2	1	2
d	2	2	1	0	1
e	3	2	2	1	2

(a): **(b):** **(c):**

Figure 3.4: Distance transform of a pair of consecutive frames (1D example). The frames consist of five possible positions (denoted a-e) and there are two trajectories passing through the frames (a). The binary image is obtained by interpreting each tracklet as a pixel. The horizontal coordinate corresponds to frame t_1, the vertical to frame t_2 (b). The distance transform of the binary image with respect to (3.9) ($\lambda_{trunc} = 2$) (c).

where $DT(.)_i$ denotes the value of distance transform for pixel i.

To define a loss function applicable to sets of trajectories we transform every frame and every pair of consecutive frames into a binary image and calculate a sum of Baddeley's losses of these images:

$$l(R, \hat{R}) = \sum_{t=1}^{n} l_{Badd}(R^{(t)}, \hat{R}^{(t)}) + \sum_{t=1}^{n-1} l_{Badd}(R^{(t,t+1)}, \hat{R}^{(t,t+1)}) \tag{3.8}$$

A single frame $R^{(t)}$ can be transformed into a binary image as follows: pixels that correspond to a position of a trajectory are considered as foreground and all other pixels as background. A pair of consecutive frames $R^{(t,t+1)}$ can be transformed into a binary image in a similar way (see Figure 3.4 for a 1D example). The pixels of the binary image represent all possible tracklets between positions in these frames (therefore the resulting image has twice as many dimensions as the single frame images). The foreground pixels are those that correspond to part of some trajectory and all other pixels are background. The distance transform of the resulting binary image is not calculated with respect to (3.6) but instead the distance of two pixels is defined as sum of truncated distances of positions of the corresponding tracklets:

$$dist(ij, i'j'; \lambda_{trunc}) = dist(i, i'; \lambda_{trunc}) + dist(j, j'; \lambda_{trunc}), \tag{3.9}$$

where i, i' are positions in frame t and j, j' are positions in frame $t+1$.

The tracklet related terms are essential for the loss function, because with single frame terms only it would be unable to distinguish between different sets of trajectories such that the objects occupy the same positions but are linked differently across the frames. On the other hand including all possible tracklets is intractable, because their number is very large even for relatively small input sequences. This can be resolved by including only a subset of tracklets: since the objects typically cannot exceed certain speed limit, it is natural to approximate the full loss using only tracklets conforming with this limit. This greatly reduces the computation time without losing ability to distinguish arbitrary sets of trajectories.

This loss function has several appealing properties. Since it does not require to explicitly match trajectories from R with trajectories from $\hat{R}$, it is naturally defined for arbitrary sets of trajectories. It is also easy to calculate, because the distance transforms can be obtained by a linear time algorithm [129].

3.2.1 Bayes Risk

In the Bayesian framework the optimal set of trajectories R^* is obtained by minimizing the expected loss

$$R^* = \operatorname*{argmin}_{\hat{R}} \mathbb{E}_{p_\theta}\left[l(R, \hat{R})\right]. \tag{3.10}$$

Since for the purpose of the inference we transform the set of trajectories into several binary images, we first describe the optimization task for a single binary image B and then generalize it to the tracking domain. The loss function for a single binary image is directly the Baddeley's loss (3.7). If we substitute the loss into (3.10) and discard terms that cannot influence the optimal solution, we end up with the following optimization task:

$$B^* = \operatorname*{argmin}_{\hat{B}} \sum_i \left(DT(\hat{B})_i^2 - 2DT(\hat{B})_i \mathbb{E}_{p_\theta}\left[DT(B)_i\right]\right). \tag{3.11}$$

In order to find the optimal solution one has to solve two difficult subproblems. One is the estimation of $\mathbb{E}_{p_\theta}\left[DT(B)_i\right]$, which in the tracking domain corresponds to the expected distance of every pixel and tracklet of length 2 to the nearest trajectory. The other subproblem is the actual optimization, i.e., finding a binary image, such that its truncated distance transform minimizes the Bayes risk given by the right-hand side of formula (3.11). Even for a single binary image this is a difficult task due to complex interdependencies among its pixels. To provide an intuition we equivalently rewrite the optimization task (3.11) as follows:

$$B^* = \operatorname*{argmin}_{\hat{B}} \sum_{j:\hat{B}_j=1} \sum_{i\in\mathcal{N}(j)} \left(DT(\hat{B})_i^2 - 2DT(\hat{B})_i \mathbb{E}_{p_\theta}\left[DT(B)_i\right]\right), \tag{3.12}$$

where $\mathcal{N}(j)$ is the influence zone of j, i.e., the set of background pixels, for which j is the closest foreground pixel. The shape and size of the influence zone depends on the truncation constant λ_{trunc} and also on values of nearby pixels. This results in huge number of possible combinations, which makes optimization of (3.11) a non-trivial task. This number is even larger in the tracking domain, because the resulting optimization task involves binary representations of multiple frames.

We estimate $\mathbb{E}_{p_\theta}\left[DT(B)_i\right]$ using an MCMC sampling algorithm and employ a greedy algorithm to minimize the risk (3.11). Both approaches are described in the following subsections.

Model Sampling

Our probabilistic model (3.1) is in fact a fully connected Conditional Random Field, where each variable represents a single trajectory. The usual procedure for Gibbs sampling is to iteratively simulate from conditional distributions of single variables. In our case these can be seen as hidden Markov chains such that the hidden variables correspond to the frames and their state space consists of all pixels and the dummy position. Although samples from a hidden Markov chain can be generated by a polynomial algorithm [150], its time complexity depends quadratically on the size of the state space, which makes it prohibitively slow for our purposes. Instead, we use Gibbs sampling in a more restricted way and at a time resample the position of a trajectory in a single frame only. This approach is tractable and enjoys the same asymptotic properties as "full" Gibbs sampling.

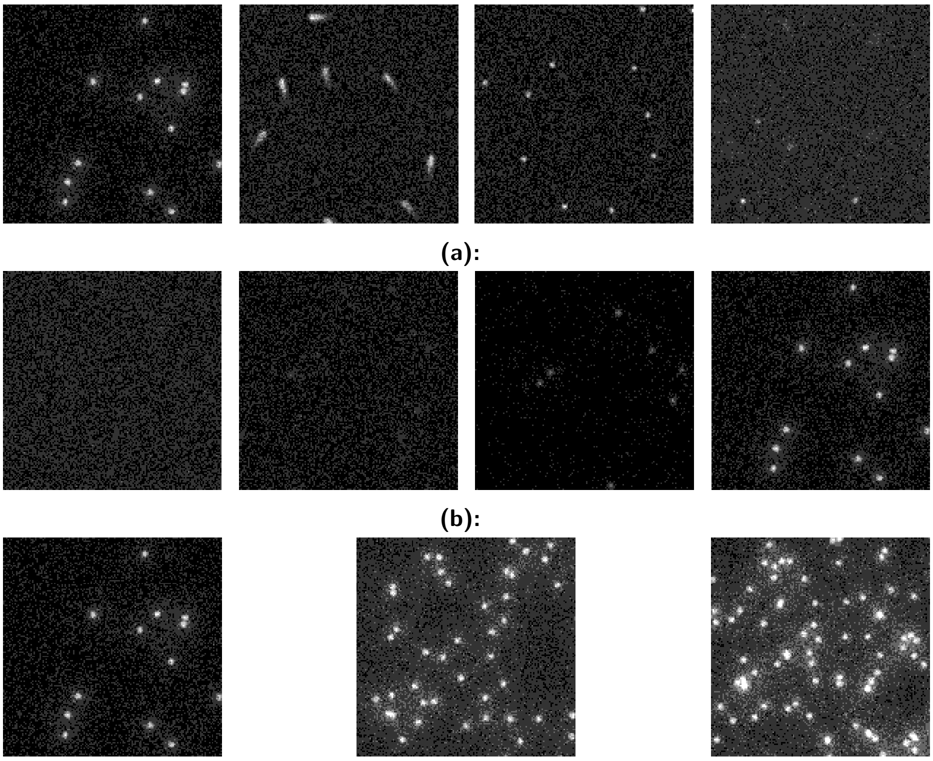
However, in our case Gibbs sampling (both full and restricted) suffers from poor mixing, so in addition we use the following Metropolis-Hastings scheme:

1. Take two trajectories and some frame t
2. Cut both trajectories in frame t and swap the sequences in subsequent frames
3. Accept the new state with probability $\min\left\{1, \frac{p_\theta(R'|M)}{p_\theta(R|M)}\right\}$, where R' denotes the set of trajectories after the cut-swap procedure was performed.

The resulting MCMC algorithm consists of iteratively repeating the restricted Gibbs and cut-swap schemes.

Risk Minimization

Formally, the final tracking result is a binary representation of a set of trajectories, such that its distance transform minimizes the Bayes risk. In order to circumvent the difficulties related to the influence zones we employ a greedy algorithm, which directly constructs the optimal set of trajectories. It starts with an empty set of trajectories and iteratively adds new trajectories and extends existing ones, such that after each step the risk decreases as much as possible. The procedure *Extend* iterates over all trajectories in R^*, that reached the $(t-1)$-th frame and for each of them calculates the change of the risk for all possible extensions to the t-th frame. The extension with the largest decrease of the risk is selected and R^* is changed accordingly. The procedure stops, when no trajectory extension can decrease the risk. The procedure *Start* iteratively adds new trajectories starting in the t-th frame, such that adding each trajectory leads to the largest possible decrease of the risk. If adding a new trajectory would increase the risk, the algorithm proceeds to the next frame. This algorithm is easy to implement but it remains open whether some optimality guarantees could be proved or whether there is a different algorithm with better properties.



(a):

(b):

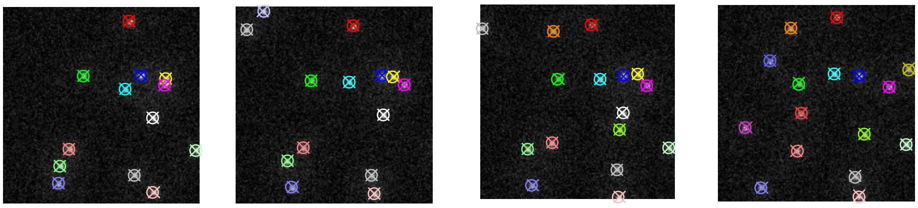
(c):

Figure 3.5: 2012 ISBI Particle Tracking Challenge Data. Four biological scenarios were simulated (from left to right): vesicles, microtubules, receptors and viruses (a). Each scenario contains images with four different SNR (from left to right, illustrated on images of vesicles): 1, 2, 4 and 7 (b). Images with three different particle densities are available for each SNR level (c). For the sake of clarity only 150 × 150 px segments of the original 512 × 512 px images are shown.

3.3 Experimental Results

3.3.1 The Data

We test our method on artificial, but highly realistic data from the 2012 ISBI Particle Tracking Challenge [135]. It contains simulated videos of fluorescence microscopy images of vesicles in the cytoplasm, microtubule transport, membrane receptors and infecting viruses (3D) with three different densities (on average 100, 500 and 1000 objects per frame) and four different SNR: 1, 2, 4 and 7 (see Figure 3.5). Motion types correspond to the dynamics of real biological particles: Brownian for vesicles, directed (near constant velocity) for microtubules and random switching between these two for receptors and viruses. The objects may appear and disappear randomly at any position and any frame. The data contain ambiguities similar to those in real data, including noise, clutter, parallel trajectories, intersecting and visual merging and splitting.



(a) : Frame 1 **(b) :** Frame 4 **(c) :** Frame 7 **(d) :** Frame 10

Figure 3.6: Tracking results for the sequence of vesicles with SNR=7 (150 × 150 px segments). Objects found by the tracker are marked by crosses and ground truth by circles.

3.3.2 Tracking Vesicles

We demonstrate the performance of our method on sequences with low density vesicles for all four levels of SNR. Each sequence consists of 100 images (512 × 512 px) with (on average) 100 objects per image. As the average lifespan of the objects is about 20 frames, every sequence contains roughly 500 objects. Vesicles appear as brighter blobs in the images but due to the noise the pixel values are not reliable detection scores. Instead we enhanced the contrast and reduced the noise by the following filter:

1. Convolve the input images with 3×3 Gaussian kernel ($\sigma = 1$) to filter out the noise

2. Choose two thresholds $0 < \lambda_{low} < \lambda_{high} < 1$, replace all the pixel values lower than λ_{low} by λ_{low} and all the pixel values higher than λ_{high} by λ_{high}

3. Normalize the pixel values to interval $[0, 1]$ and use them as detection scores

Pixel values of the filtered images were then taken as detection scores. In all four experiments we used the same model parameters and only the detector thresholds λ_{low} and λ_{high} were tuned using the first image of the sequence. Maximum velocity of the objects was bounded to 15 pixels per frame.

To make our method comparable to the other contributions to the 2012 ISBI Particle Tracking Challenge, we use the primary evaluation objectives from the challenge. The true positive rate $\alpha(R^*, R_{GT})$ indicates the percentage of trajectories from the ground truth R_{GT} that can be matched with some trajectory from R^*. It takes values from $[0, 1]$ such that 1 means a perfect match and 0 indicates, that no valid match can be found. Since this objective does not account for false positives, we use in addition the false positive (FP) rate $\beta(R^*, R_{GT}) \in [0, \alpha(R^*, R_{GT})]$. If the FP rate attains its maximum value, there are no false positive trajectories in R^*. The other extreme is $\beta(R^*, R_{GT}) = 0$, which indicates that R^* consists of infinite number of trajectories and none of them can be matched with a ground truth counterpart. Finally, we report the root mean square error (RMSE) of positions of true positives. Instead of creating a single measure of quality, these objectives

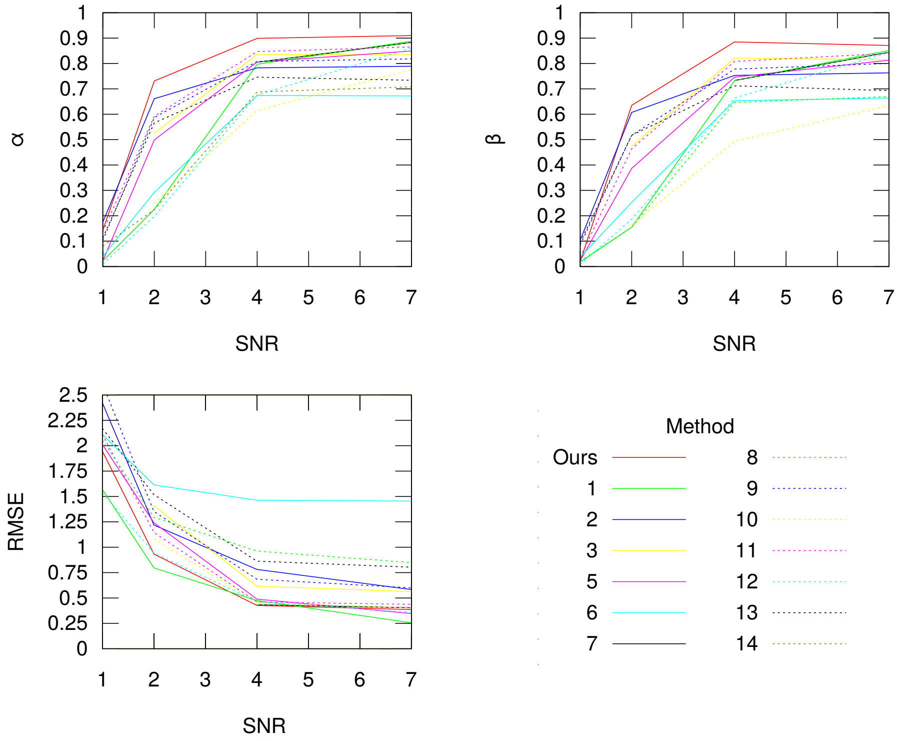


Figure 3.7: Objectives α, β and RMSE from the 2012 ISBI Particle Tracking Challenge for tracking low-density vesicles (SNR 1, 2, 4, and 7). Competing methods are numbered according to [135]. Team 4 did not submit results for low density vesicles.

are used separately in the challenge and therefore a direct comparison of two methods is possible only if one outperforms the other in all of them. For exact definition of these objectives and description of other auxiliary measures we refer the reader to [135].

Sample tracking results are shown in Figure 3.6. As shown in Figure 3.7, for SNR 2, 4 and 7 our method outperforms all the methods from the challenge in objectives α and β and it is highly competitive in RMSE. Similarly to the other methods, it performs poorly on the data with SNR 1. This is caused by an oversimplified detector, which fails in that case beyond the model's ability to recover. We believe that better results could be achieved by incorporating the tracker and an improved detector into a joint model.

3.4 Summary

In this chapter we proposed a probabilistic graphical model for tracking dimensionless objects, which integrates submodels for appearance, motion and object interaction. The input to the model is not a fixed set of detection candidates but it instead uses a detection map (pixel-wise detection scores) and determines the number and positions of objects by itself. The final tracking result is obtained by minimizing the Bayes risk with a loss function based on Baddeley's delta loss for binary images. The method was tested on data from the 2012 ISBI Particle Tracking Challenge and proved to be competitive with state-of-the-art methods.

Chapter 4

Probabilistic Shape Prior

Markov Random Fields have a long history as prior models for image segmentation. In their seminal work Geman and Geman [65] proposed to use the Ising model as a segmentation prior. The whole model is composed of two layers. The variables on the image level correspond to the image pixels and their label is fixed to the pixel intensity. The segmentation layer is composed of binary variables (one variable per pixel) that represent the segmentation and are connected by an edge to an image variable corresponding to the same pixel and to neighboring segmentation variables. The potential functions associated with the edges in the segmentation layer serve as soft constraints, which prefer neighboring pixels to have the same label. In the simplest case the potentials are translation invariant and their strength is the only free parameter of the model.

Thanks to its simple structure and intuitive properties this model quickly became widely used for segmentation tasks. Its structure is however also responsible for its main limitation – since the edges connect only pairs of pixels, it is unable to represent more complex shape assumptions, e.g. smooth boundaries, sharp tips or even global shapes. To achieve this capability it is necessary to model interactions of larger groups of pixels. Naively introducing higher order cliques would however make the model intractable, because number of parameters of each clique grows exponentially with its size. A better way is to enhance the model by another layer of variables that are connected by edges with multiple segmentation variables. It can be shown, that marginalizing over the additional layer results in a MRF with higher order potentials and this structure is therefore suitable for modeling more complex shapes.

The shape assumptions represented by a model of this class are not hard coded by its structure but can be learned from the training data. It is reasonable to assume, that training data for the segmentation layer are available, but this is not the case for the additional layer – the labelling of the additional layer is in fact a representation of shapes in the segmentation layer, which has to be learned first and therefore cannot be available beforehand. Consequently, the model parameters must be learned in unsupervised way.

In this chapter we propose a segmentation shape prior based on MRFs on bipartite graphs and an algorithm for unsupervised parameter learning. The algorithm is a modified EM algorithm, which replaces the likelihood estimator in the M step by the pseudolikelihood. Although it can be used for arbitrary MRFs, it is mainly intended for MRFs on bipartite graphs since it can take advantage of their structure. The model is used as a generic shape prior for circular cells and for segmentation of lungs in X-ray chest radiographs.

4.1 The Model

MRFs on bipartite graphs can be described as follows. Let (V, E) be an undirected bipartite graph and V_1, V_2 denote its parts. Let X be a collection of K_1-valued random variables indexed by vertices of V_1. That is, $X = \{X_i \mid i \in V_1\}$, where each X_i is a K_1-valued random variable. Similarly, Y denotes a collection of K_2-valued random variables indexed by vertices of the second part V_2. Both co-domains K_1 and K_2 are assumed finite. We denote realizations of the random field (X, Y) by (x, y), i.e.,

$$x\colon V_1 \to K_1, \quad y\colon V_2 \to K_2.$$

The joint probability distribution function of an MRF on (V, E) can be written as an exponential family (assuming strictly positive probability mass)

$$p_\theta(x, y) = \frac{1}{Z(\theta)} \exp \sum_{ij \in E} \langle \boldsymbol{\zeta}(x_i, y_j), \boldsymbol{\theta}_{ij} \rangle, \tag{4.1}$$

where $\theta = \{\boldsymbol{\theta}_{ij} \mid ij \in E\}$ denotes the set of model parameters and $\boldsymbol{\zeta}\colon K_1 \times K_2 \to \{0, 1\}^{|K_1||K_2|}$ is a vector valued indicator function. The output vector of $\boldsymbol{\zeta}(x_i, y_j)$ is composed of 0 except for the element which corresponds to the actual labelling of x_i and y_j and is equal to 1. This allows to use the dot product to "select" the corresponding parameter from the parameter vector $\boldsymbol{\theta}_{ij}$.

This model class includes Restricted Boltzmann Machines [79], which are often used in the context of deep learning [16]. An RBM in its narrow sense assumes that the co-domains of both groups of random variables are binary $|K_1| = |K_2| = 2$ and the bipartite model graph is complete.

Despite the fact that the considered model class has pairwise factors only, it can be used to model higher order factors in the following way. If the variables X_i, $i \in V_1$ are considered as “visible” and the variables Y_j, $j \in V_2$ as latent, then, by marginalizing over the field Y, we get a Markov Random Field with higher order factors for the field X.

Notice that due to the bipartiteness of the graph both conditional probability distributions $p_\theta(y \mid x)$ and $p_\theta(x \mid y)$ factorize

$$p_\theta(y \mid x) = \prod_{j \in V_2} p_\theta(y_j \mid x_{\mathcal{N}(j)}), \tag{4.2}$$

where $\mathcal{N}(j) = \{i \in V_1 \mid ij \in E\}$ denotes the neighborhood of the vertex $j \in V_2$.

4.2 Parameter Learning

We assume from here on that the variables X_i, $i \in V_1$ are visible, whereas the variables Y_j, $j \in V_2$ are latent and consider the task of parameter estimation given an i.i.d. sample $\mathcal{T}$ of $|\mathcal{T}|$ realizations of the field X. It is assumed that the realizations were generated by $p_\theta(x) = \sum_{y \in \mathcal{Y}} p_\theta(x,y)$ with unknown θ. If the maximum likelihood estimator is used, the task is

$$\frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \log \sum_{y \in \mathcal{Y}} p_\theta(x,y) \to \max_\theta, \tag{4.3}$$

where $\mathcal{Y}$ denotes the set of all possible realizations of the field Y. Substituting the model class (4.1), the task reads

$$L(\theta) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \log \sum_{y \in \mathcal{Y}} \exp \sum_{ij \in E} \langle \boldsymbol{\zeta}(x_i, y_j), \boldsymbol{\theta}_{ij} \rangle - \log Z(\theta) \to \max_\theta \tag{4.4}$$

where

$$Z(\theta) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \exp \sum_{ij \in E} \langle \boldsymbol{\zeta}(x_i, y_j), \boldsymbol{\theta}_{ij} \rangle \tag{4.5}$$

denotes the partition sum. It can be shown that both terms in (4.4) are convex functions of θ. The log-likelihood $L(\theta)$ is therefore a difference of convex functions.

4.2.1 Discussion of existing methods

The gradient of the log-likelihood is easy to derive

$$\nabla_{\boldsymbol{\theta}_{ij}} L(\theta) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \mathbb{E}_\theta(\boldsymbol{\Phi}_{ij} | X = x) - \mathbb{E}_\theta(\boldsymbol{\Phi}_{ij}), \tag{4.6}$$

where $\boldsymbol{\Phi}_{ij}$ denotes the random variable $\boldsymbol{\Phi}_{ij}(X,Y) = \boldsymbol{\zeta}(X_i, Y_j)$, $ij \in E$. The first term in (4.6) is tractable because the conditional probability distribution $p_\theta(y \mid x)$ factorizes, which makes the computation of the conditional expectations tractable, and because the sum over the elements of the learning sample $\mathcal{T}$ is tractable. The second term is, on the contrary, not tractable – it requires to compute pairwise marginal probabilities $p_\theta(x_i, y_j)$. It is well known, that calculating the marginals for an MRF is #P hard [27]. Therefore, one has to rely on approximate algorithms. Let us shortly discuss possible options.

Variational methods like belief propagation or other message passing algorithms fail to estimate pairwise marginal statistics even approximately [78]. This can be explained by the following argument. All these methods approximate the pairwise log-marginals by

$$\log p(x_i, y_j) \sim a_i(x_i) + \theta_{ij}(x_i, y_j) + b_j(y_j), \tag{4.7}$$

i.e., as being equal to $\boldsymbol{\theta}_{ij}$ up to a modular function. While this is true for trees, it is wrong for general graphs because correlations caused by loops are ignored.

Another option for estimating the required marginals is Gibbs sampling. However, Gibbs sampling is very slow if applied correctly [181]. To generate just one realization (x, y), it is often necessary to run thousands of iterations of the sampler.

A third option is a *stochastic gradient* method which is often used in the context of RBMs and is designated as Persistent Contrastive Divergence (PCD) [188, 189]. PCD keeps a realization $(x^{(a)}, y^{(a)})$ at each iteration a. The current model estimate $\theta^{(a)}$ is used to resample the realization $(x^{(a+1)}, y^{(a+1)})$ The new realization is then used to estimate the second term of the gradient, simply by replacing the expectation of $\mathbf{\Phi}_{ij}$ by its realization $\boldsymbol{\zeta}(x_i, y_j)$. Finally, a new model estimate $\theta^{(a+1)}$ is obtained by applying a gradient step. Clearly, there are no guarantees for convergence to the global optimum because the objective function is not concave and the true gradient is replaced by an approximation.

We may try to avoid to deal with $L(\theta)$ directly by applying the EM algorithm. An iteration of it reads as follows.
E-step: Calculate posterior probabilities

$$\beta^{(a)}(y \mid x) := p_{\theta^{(a)}}(y \mid x) \tag{4.8}$$

for each realization $x \in \mathcal{T}$ using the current parameter estimate $\theta^{(a)}$. This task is feasible for the considered model class (see (4.2)).
M-step: Given the current $\beta^{(a)}$ maximize the log-likelihood for complete information

$$L_c(\theta) = \frac{1}{|\mathcal{T}|} \sum_{x \in \mathcal{T}} \sum_{y \in \mathcal{Y}} \beta(y \mid x) \log p_\theta(x, y) \to \max_\theta . \tag{4.9}$$

Let us denote by p^* the distribution $p^*(x, y) = \beta(y \mid x) p^*(x)$, where $p^*(x)$ is the empirical distribution associated with the sample $\mathcal{T}$. Substituting the model (4.1), the objective function in the M-step can be written as

$$L_c(\theta) = \sum_{ij \in E} \langle \mathbb{E}_{p^*}(\mathbf{\Phi}_{ij}), \boldsymbol{\theta}_{ij} \rangle - \log Z(\theta). \tag{4.10}$$

It is concave in θ, but, again, the problem is the gradient of the second term (the logarithm of the partition sum Z). Computing its components requires to compute pairwise marginal statistics of the model $p_\theta(x, y)$ and is therefore not tractable.

4.2.2 A Modified EM Algorithm

Following the interpretation given by one of the authors of the EM-algorithm [169], the task to be solved in each M-step is itself a (parameter) learning task, now in presence of complete data. The model parameters θ must be estimated given the "observed" distribution $p^*(x, y)$. As we have seen, this task is still not tractable for the considered class of MRFs. On the other hand, the definition of p^* implies that i.i.d. samples from p^* can be easily generated.

The key idea is therefore to replace the maximum likelihood estimator in the M-step by any consistent *and* tractable estimator. A reasonable choice is the pseudolikelihood estimator.

Let us denote by $\mathcal{T}^*$ an i.i.d. sample of realizations (x, y) generated from $p^*(x, y)$. The pseudolikelihood estimator for MRFs on bipartite graphs reads

$$L_p(\theta) = \sum_{(x,y)\in\mathcal{T}^*} \Big[\log p_\theta(y \mid x) + \log p_\theta(x \mid y)\Big] \to \max_\theta . \tag{4.11}$$

The objective function is concave and has a tractable gradient

$$\nabla_{\boldsymbol{\theta}_{ij}} L_p(\theta) = \sum_{(x,y)\in\mathcal{T}^*} \Big[2\boldsymbol{\zeta}(x_i, y_j) - \mathbb{E}_\theta(\boldsymbol{\Phi}_{ij}|X = x) - \mathbb{E}_\theta(\boldsymbol{\Phi}_{ij}|Y = y)\Big]. \tag{4.12}$$

Summarizing, each iteration of the modified EM algorithm reads as follows
E-step: Calculate posterior probabilities

$$\beta^{(a)}(y \mid x) := p_{\theta^{(a)}}(y \mid x) \tag{4.13}$$

for each realization $x \in \mathcal{T}$ using the current parameter estimate $\theta^{(a)}$. Sample one (or several) realizations y for each $x \in \mathcal{T}$. These data define the current sample $\mathcal{T}^*$ for the M-step.
M-step: Maximize the pseudolikelihood

$$L_p(\theta) = \sum_{(x,y)\in\mathcal{T}^*} \Big[\log p_\theta(y \mid x) + \log p_\theta(x \mid y)\Big] \tag{4.14}$$

e.g. by using a gradient ascend algorithm. Set $\theta^{(t+1)}$ to be equal to the maximizer.

It remains to discuss the choice for the initial model parameters $\theta^{(0)}$. The simplest option is to choose them randomly in the vicinity of the origin. Yet there is a better option for MRFs on bipartite graphs. Let us consider the sub-graph defined by a vertex $j \in V_2$ and its neighbors $\mathcal{N}(j) \subset V_1$ and the random variables Y_j, X_i, $i \in \mathcal{N}_j$. Taken alone, they define a naive Bayes model. The parameters of such a model can be learned by a standard EM-algorithm. Applying it for each of the sub-models separately, gives a good initialization for the model parameters.

In summary, the resulting double loop algorithm is easy to implement and has the same per iteration time complexity as PCD. On the other hand, we have no proof that the sequence of likelihood values $L(\theta^{(a)})$ is increasing. This should be true in the limit of an infinite training sample because the pseudolikelihood estimator is known to be consistent. However, there is no such guarantee for finite training samples. We will compare the proposed algorithm with PCD for direct likelihood maximization in the experimental section.

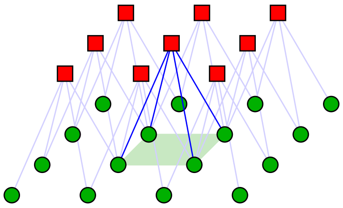


Figure 4.1: MRF on a translational invariant bipartite graph. Visible variables depicted as green circles, latent variables depicted as red squares. Edges and receptive field are highlighted for one of the latent variables.

4.3 Experiments

We aim to apply the discussed type of MRFs for shape modeling. By this we mean to model simple shapes and spatial relations (like “above, ”inside“, etc.) for segments. Bearing in mind such applications, we make the following assumptions for all presented experiments. The vertex sets V_1 and V_2 are congruent subsets of $\mathbb{Z}^2$ and the values of the random variables x_i represent segment labels. The graph structure is translation invariant, i.e., $\mathcal{N}(j+h) = \mathcal{N}(j) + h$ for all $j, h \in \mathbb{Z}^2$ such that $j, j+h \in V_2$ (see Figure 4.1). We call $\mathcal{N}(j)$ receptive field of the latent variable Y_j. The model parameters are translation invariant as well

$$\boldsymbol{\theta}_{ij} = \boldsymbol{\theta}_h, \quad \forall \{i,j\} \in E \text{ s.t. } i - j = h. \tag{4.15}$$

Please notice that the models we are using here for experiments differ from those usually used for experiments on RBMs (see e.g. [122]) in two respects. We use large size fields in contrast to usually used models of relatively small size. The latent variables are often considered as features for subsequent classification. Here in contrast, they are used to model complex distributions for the field X.

4.3.1 The Pn Model

We consider the Pn model for binary segmentations in the first experiment. It is a generalized Potts model on cliques of size λ_n [102]. The factors of the Markov Random Field associated with the cliques $\mathcal{N}(j)$ are two-valued; a large value is assigned to homogeneous realizations of $X_{\mathcal{N}(j)}$ with either of the two possible segment labels. A small value is assigned to all other realizations of $X_{\mathcal{N}(j)}$. To express this higher order model by an MRF on a bipartite graph, we make each clique $\mathcal{N}(j)$ a receptive field of a three-valued latent variable Y_j. The conditional probability distributions $p(x_{\mathcal{N}(j)} \mid y_j)$ for the first two values of Y_j are non-zero only for the two homogeneous realizations $x_{\mathcal{N}(j)} \equiv 0, 1$ respectively. The conditional probability distribution for the third value of Y_j is uniform. A mixture of the three probability distributions corresponds to a factor of the Pn model.

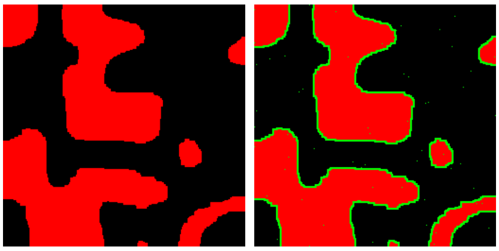


Figure 4.2: A realization (x, y) generated by the P9 model.

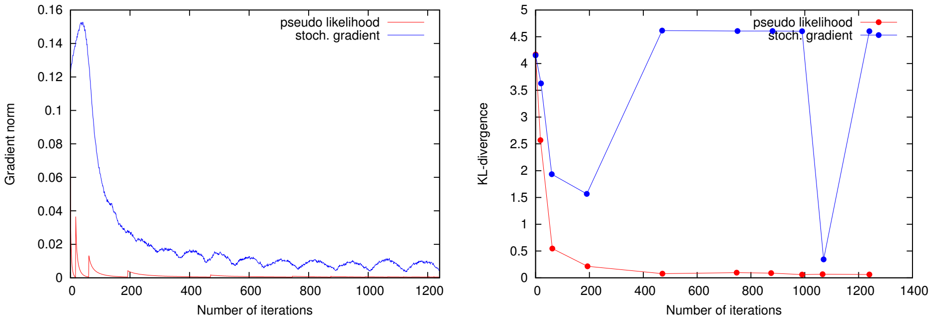


Figure 4.3: Comparison of stochastic gradient method and the proposed modified EM-algorithm. Left: norm of the gradient. Right: KL divergence from true marginals

We have implemented a $P9$ model with receptive fields of size 3×3. Figure 4.2 shows a random realization (x, y) (color coded) generated by this model. We have generated 50 realizations of X (size 256x256) by extensive Gibbs sampling (10^4 sampling iterations per example) and used them for learning. The model was learned by the stochastic gradient method (PCD) and by the proposed modified EM-algorithm. In this experiment we were not using the ”naive Bayes“-based initialization (see subsection 4.2.2). We have chosen the size 512x512 for the realization (x, y) needed for the gradient estimation in the PCD algorithm. The optimal step width for the gradient ascends were chosen empirically for each of the algorithms.

To compare the two learning algorithms, we display the L_∞ norm of the gradients over the iteration number in Figure 4.3. The sawtooth-like shape of the curve for the modified EM-algorithm is explained as follows. The (negative) pseudolikelihood and its gradient decrease in the inner loop of the algorithm (M-step). Then the y-fields are resampled using the new model estimate (E-step), what causes the jump in the gradient of the pseudolikelihood.

Overall, it is clearly seen that the proposed modified EM-algorithm converges faster by an order of magnitude and much more stable than the stochastic gradient algorithm. Of course, this comparison alone does not say anything about the models learned by the respective algorithm. The objective functions are different and, moreover, the gradient of the likelihood (in the PCD algorithm) is determined approximately only.

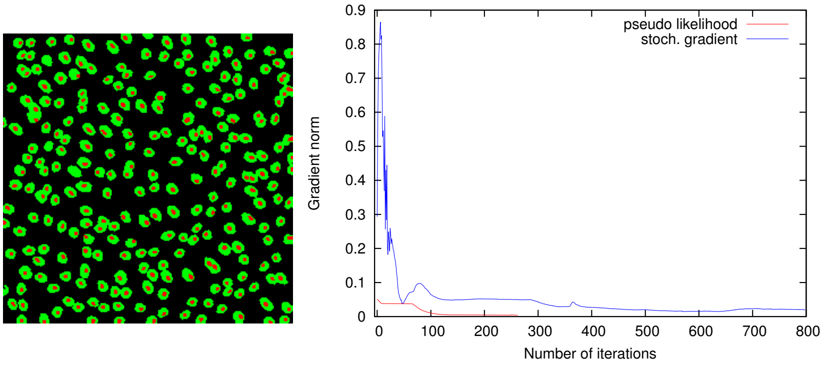


Figure 4.4: Left: cell segmentation (artificial). Right: Comparison of stochastic gradient method and the proposed modified EM-algorithm.

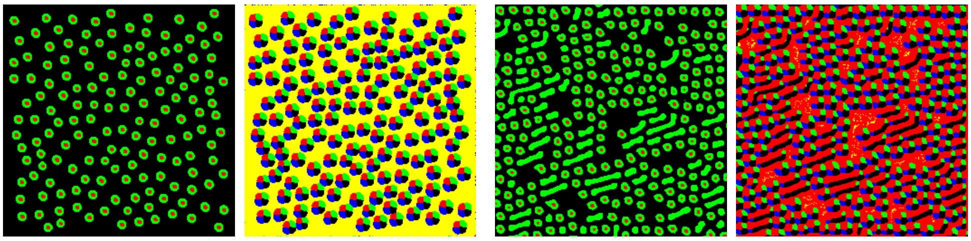


Figure 4.5: Realizations (x, y) randomly generated by the learned cell segmentation models. Left pair: model learned by modified EM-algorithm. Right pair: model learned by stochastic gradient algorithm. Three colors (red, green, black) were used to represent the possible values of x_i, $i \in V_1$ and four colors (red, green, black, yellow) were used to represent the possible values of the latent variables y_j, $j \in V_2$.

It would not be very reasonable to compare the learned models by comparing their parameters θ directly. They are not unique due to possible reparameterizations. Moreover, models with different distributions $p_\theta(x, y)$ may have the same distribution $p_\theta(x)$. Therefore we have chosen to compare the resulting marginal distributions $p_\theta(x_{\mathcal{N}(j)})$ for the receptive fields of size 3x3 which have 512 possible realizations. They were estimated for the true model as well as for each of the learned models by extensive sampling. Figure 4.3 shows the KL-divergence between the marginals of the true model and the learned models for some iteration numbers. Again, it is clearly seen that the proposed modified EM-algorithm converges faster and much more stable than the PCD algorithm.

4.3.2 Cell Segmentation

We consider a more complex model for the second experiment. The goal is to learn a prior model for segmenting cells in microscope images. We assume a typical segmentation to contain non-occluding cells with roughly circular shaped cytoplasm and circular shaped nuclei. Artificial segmentations of

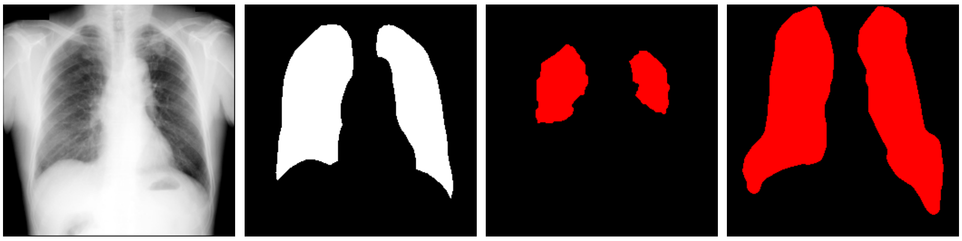


Figure 4.6: From left to right: Chest radiograph, ground truth segmentation of lung, GrabCut segmentation, smooth boundary + atlas model segmentation.

the type shown in Figure 4.4 were used as training data. To learn such segmentations we have chosen a model with the following structure. The co-domain K_1 of the variables X_i has three values corresponding to the three possible segment labels – background, cytoplasm, nucleus. The co-domain K_2 of the latent variables was chosen to have five values. The receptive fields for the latter were chosen to have roughly the size of a cell, 11×11 pixels in our case. To speed up learning, we used the "naive Bayes"-initialization (see subsection 4.2.2).

Figure 4.4 shows the learning curves for the modified EM-algorithm and the stochastic gradient algorithm. Again, the former converges much faster and more stable than the latter. Moreover, comparing realizations generated by the learned models (see Figure 4.5), it is seen that the model learned by the modified EM-algorithm generates desired segmentations after 260 learning cycles. The model learned by the PCD algorithm has not yet fully "captured" the desired segmentations even after 800 learning cycles.

4.3.3 Lung Segmentation

The aim of the last experiment differs from those of the previous experiments – here we want to demonstrate the usefulness of MRFs on bipartite graphs for segmentation tasks. Let us consider lung segmentation in X-ray chest radiographs as an example. As typical for such tasks, it is desirable to have a segmentation model which prefers e.g. smooth boundaries and simultaneously utilizes a probabilistic anatomical atlas. This is easy to achieve by using models of the considered type. A translational invariant model as shown in Figure 4.1 is extended by one more latent variable with edges to all pixels of the segmentation. This "global" latent variable realizes a "mixture" of anatomical atlases jointly with the other latent variables, which model translational invariant local segment/boundary features.

Such a model was used as a prior model for segmenting lungs in X-ray chest radiographs from the database provided by Japanese Society of Radiological Technology [177] (see Figure 4.6). The dataset consists of 247 fully annotated 12 bit images (2018 × 2048 px). The "local" latent variables were chosen to have a co-domain K_2 with 18 possible values and receptive fields of size 7×7 pixels in one case and 9×9 pixels in the other one. We also considered different component numbers (4 and 12) for the global latent variable. The models were learned on 124 randomly chosen ground truth segmentations from the database. We used the ”naive Bayes“ initialization to speed up the learning. The appearance model was chosen to be conditionally pixel-wise independent given the segmentation. The gray-value distributions for the two segment labels are assumed as mixtures of (three) Gaussians each and were learned semi-supervised for each test image (the remaining 123 images from the database) separately. For this, the segmentation was fixed in regions for which the learned atlas mixture predicts a unique a-priory decision. Slightly bigger regions (80% sure decision of the atlas mixture) were used for learning the initial appearance model.

We have used the standard GrabCut method [161] as baseline. Notice, that the underlying model is an MRF on a lattice without latent variables. The parameters of the appearance model were learned semi-supervised by fixing the segmentation in the same ”unique decision“ regions. Table 4.1 shows the average segmentation precision and its variance obtained by the models with different receptive fields and different number of labels for the global latent variable.

	GC	7x7/4	7x7/12	9x9/4	9x9/12
mean	0.521	0.822	0.836	0.829	0.839
var.	0.117	0.072	0.068	0.073	0.067

Table 4.1: Lung segmentation precision (dice metric)

It is clearly seen that the considered model class outperforms GrabCut substantially. Not surprisingly, the results are the better the bigger the receptive fields of the local latent variables (responsible for smooth boundaries) and the larger the co-domain of the global latent variable (responsible for the anatomical atlas).

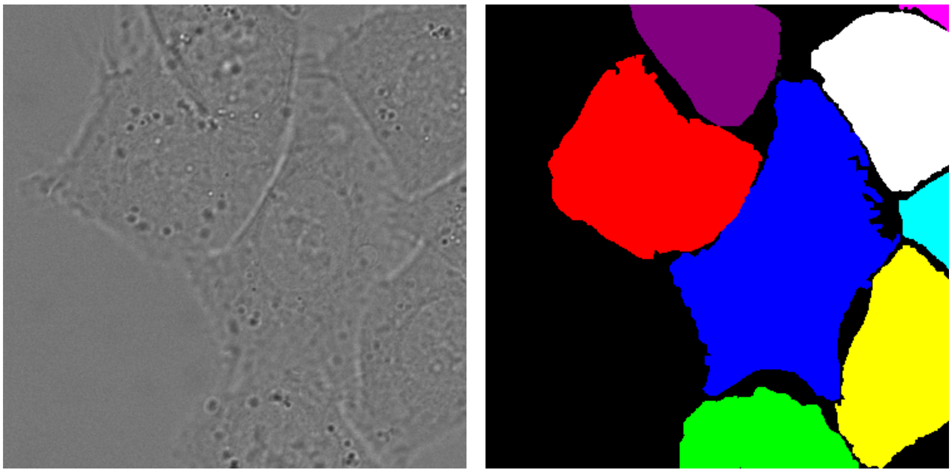
4.4 Summary

In this chapter we proposed a probabilistic shape prior for semantic segmentation. The model is an MRF on a bipartite graph. Furthermore we proposed an algorithm for unsupervised parameter learning of MRFs, which can take advantage of the special structure of the considered model class. The algorithm is a modified EM algorithm, which replaces the likelihood estimator in the M step by the pseudolikelihood. The model was used as a generic shape prior for circular cells and for segmentation of lungs in X-ray chest radiographs, where it outperformed the GrabCut baseline. The learning algorithm was compared with the Persistent Contrastive Divergence and in the performed experiments was shown to be more stable and to converge faster.

Chapter 5

Joint Segmentation, Detection and Tracking

Performance of tracking-by-detection methods depends heavily on the quality of the detector. If the detector misses an object or estimates its shape incorrectly, the data association module has no way to correct the error, because it cannot generate detection candidates on its own. This problem is typically tackled by generating an overcomplete set of detection candidates, which decreases the possibility of the aforementioned errors. Another option is to replace the underperforming detector by a better one, which can be done easily in the tracking-by-detection framework.



(a) : Input image

(b) : Instance segmentation

Figure 5.1: Motivation for integrating detector and tracker into a joint model (HeLa cells on a flat glass). Instance segmentation of individual cells using the image alone is difficult due to poorly visible boundaries, but the detector can benefit from exploiting their poses in neighboring frames.

Although these approaches are effective in many cases, they inherently fail in situations, when the image alone does not provide enough information to correct the error. Consider for example Figure 5.1, which shows HeLa cells on a flat glass. It is very hard to estimate the shapes of individual cells accurately, because their boundaries are in many places indistinguishable

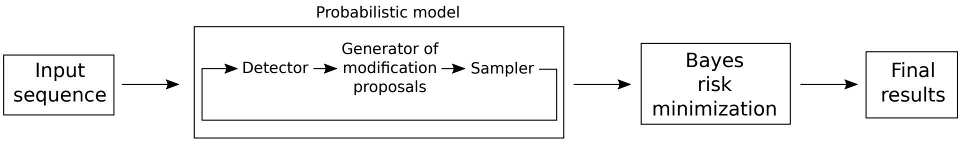


Figure 5.2: Workflow of the method. The input data are first processed by the detection module, which generates the initial set of detection candidates. Several samples are then drawn from a probabilistic tracking model by a MCMC algorithm. The model integrates the detector and the data association module using a feedback loop, which allows to revisit the initial set of detection candidates. The final results are obtained by minimizing the Bayes risk.

from their interior. Furthermore the cells do not have any specific global shape (except for being roughly elliptic) and their dimensions vary greatly, which mitigates potential benefits of a global shape prior.

Ambiguities similar to Figure 5.1 can be sometimes resolved by considering states of the objects in neighboring frames, where the boundaries are better visible (e.g. because the objects drifted from each other). This can be achieved by integrating the detector and the data association module into one joint model.

In this chapter we propose a probabilistic model for joint segmentation, detection and tracking. The workflow of the whole method is shown in Figure 5.2. The detection and data association modules are integrated into a single probabilistic model defined implicitly in terms of an MCMC algorithm. The integration is achieved by a feedback mechanism, that allows the algorithm to dynamically alter parameters of the detector and create new detection candidates in order to correct detection errors.

The final tracking result is obtained by minimizing the expected loss (Bayes risk). The inference task involves certain marginal statistics of the model that are difficult to calculate exactly, but can be estimated from the samples generated by the MCMC algorithm. Thus, our method first generates several samples (each representing a set of trajectories of moving objects and their parent-offspring relations) and then uses them to obtain the final result.

As our method is able to estimate shapes of the tracked objects, we represent the final result as a segmentation of pixels of the input sequence (we denote it as pixel-level representation). However, for practical implementation it is more convenient to define the MCMC algorithm in terms of higher order objects e.g. detection candidates or tracklets (object-level representation). Throughout this chapter we use both representations – object-level for the MCMC algorithm and pixel-level for the inference and parameter learning. We define both representations such that the object-level representation can be transformed into an equivalent pixel-level one using a simple mapping and consequently a probabilistic model defined for object-level representations can be applied to pixel-level as well.

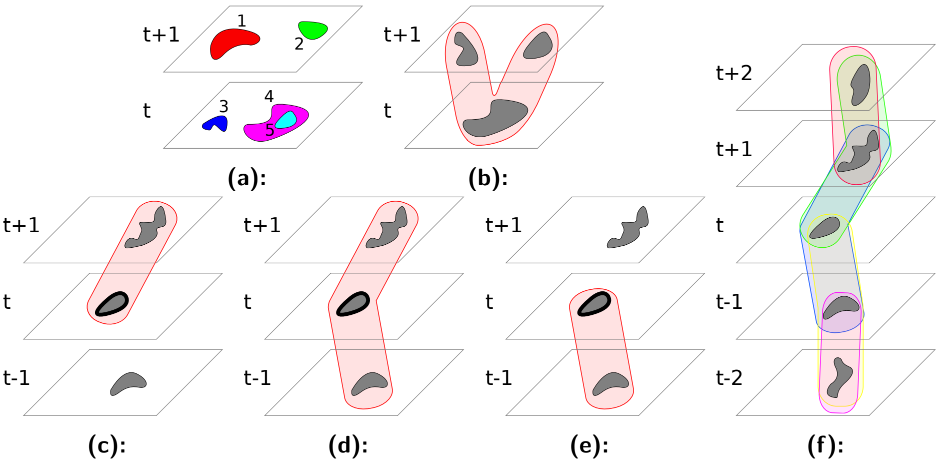


Figure 5.3: Object-level variables. Detection candidates (numbered 1-5) (a). A cell division variable (represented by the red blob) linking a parent detection candidate in frame t with offsprings in frame t+1 (b). Tracklets linking a reference detection candidate in frame t with detection candidates in the surrounding frames (c)-(e). A trajectory composed of five tracklets (f). Frame numbers are indicated by $(t-2)..(t+2)$.

5.1 The Model

The model is defined implicitly in terms of an MCMC algorithm, which repeatedly sweeps through the input sequence I and constructs a tracking sample (i.e., a set of trajectories of moving objects, their shapes and parent-offspring relations) in a series of local modifications. Every modification is selected from a number of modification proposals of various types, e.g. start new trajectory, modify shape of an existing trajectory in certain frame, merge two trajectories etc. The sample is represented as a labeling of binary variables, each representing a small part of the tracking result, e.g. an object in one frame, tracklet, etc. (thus the object-level representation). The tracking result is composed of variables labeled by 1. We use variables of three types:

1. **Detection variables**. Every detection variable represents a single detection candidate, i.e., a set of pixels, that corresponds to a shape of one object in one frame. In general there are no restrictions imposed on the shape of the detection candidates, they may contain holes or even be composed of multiple non-connected components. There might be multiple detection candidates with different shapes corresponding to single object and detection candidates corresponding to different objects may overlap with each other. Number of detection variables is not fixed because new candidates might be created by the feedback mechanism. (Figure 5.3a)

2. **Tracklets**. Every tracklet variable links a detection candidate (for convenience we call it the reference detection candidate of the tracklet) with

other candidates in the following and/or previous frame(s). Distinguishing the reference and other detection candidates simplifies definition of the motion submodel – the deviation of the tracklet from the motion assumption can be quantified as the difference between the position of the reference candidate and the position predicted using the remaining candidates. The maximum length of tracklets is dictated by the motion assumption. For example, Brownian motion assumption limits tracklets to one frame to the future and one to the past, constant velocity allows two frames in both directions etc. In order to represent motion of objects near the beginning and the end of their trajectories the model also contains tracklets shorter than the maximum length given by the motion assumption. There is a tracklet variable for every tuple of detection candidates, that does not violate any domain specific constraints, e.g. a maximum distance an object can travel between two frames (Figures 5.3c-5.3e). Longer trajectories are composed of several chained tracklet variables (Figure 5.3f).

3. **Cell division events**. Cell division variables establish parent-offspring relationships for a triple of detection candidates. There is a cell division variable for every triple that fulfills the following conditions: both offsprings are in a frame successive to the parent and their (spatial) distance from the parent is smaller than some user defined threshold (Figure 5.3b).

5.1.1 Feasibility Constraints

Not every labeling of object-level variables can be uniquely interpreted as a tracking result. For example setting value of a tracklet variable to 1 and in the same time values of linked detection variables to 0 leads to an ambiguity, because it is not clear, whether this configuration corresponds to a trajectory of a moving object (as suggested by the tracklet variable) or empty space. To make sure that the MCMC algorithm generates interpretable labelings only we assign zero probability to all labelings, that do not fulfill the following constraints:
Constraints for the detection variables:

- A detection variable can be labeled by 1 only if it does not overlap with another detection variable labeled by 1.

Tracklets & Trajectories:

- Tracklets can be labeled by 1 only if all detection candidates they link are labeled by 1.

- If two tracklet variables share detection candidates, they can be labeled by 1 only if all non-shared detection candidates belong to different frames. This constraint prevents representing a part of a trajectory by multiple redundant tracklets.

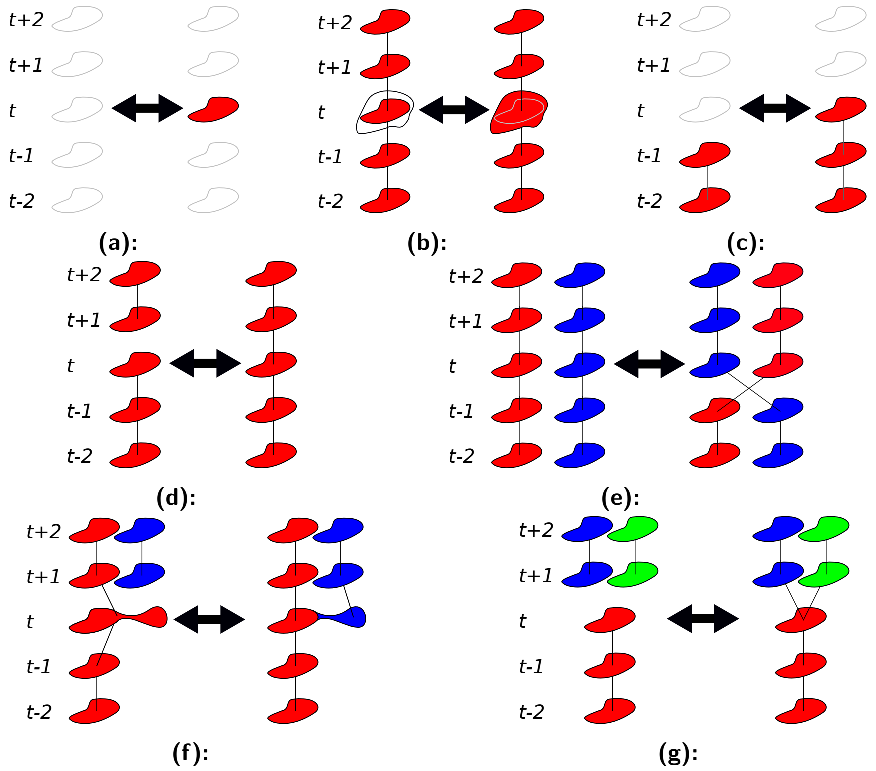


Figure 5.4: Modification proposals. Add/remove single detection candidate (a). Modify shape of a trajectory in one frame (b). Prolong/shorten trajectory by one frame (c). Append two trajectories; cut a trajectory into two pieces (d). Swap proposal (e). Split/merge proposal (f). Establish/cancel a cell division event (g). Empty shapes represent detection variables labeled by 0 whereas filled shapes labeled by 1. Frame numbers are indicated by $(t-2)..(t+2)$.

- If two tracklet variables share detection candidates, they can be labeled by 1 only if their reference candidates belong to different frames. This constraint ensures that the motion submodel is evaluated for every detection candidate in a trajectory only once. This is not guaranteed by the previous constraint, because the beginning and the end of the trajectory is represented by shorter tracklets and without this constraints they could share the reference detection candidate with a longer tracklet.

- Trajectory of length l frames is composed of l tracklets of maximum possible length.

Cell divisions:

- A cell division variable can be labeled by 1 only if its parent detection candidate is the last frame of some trajectory and its offspring candidates are the first frames of different trajectories.

5.1.2 Modification Proposals

In every iteration the MCMC algorithm selects a modification proposal that alters a small part of the tracking sample. The proposals are of various types. There might be multiple possible proposals of certain type (e.g. for changing the shape of an object), but in some cases no proposals of certain type are valid, i.e., they would lead to a labeling that violates the feasibility constraints.

The simplest modification proposals involve changing label of a single detection variable. Since every detection candidate occupies only one frame, we indicate it for convenience in the notation – $\mathit{AddRemove}(t)$ denotes the set of all modification proposals that change the label of a single detection variable occupying frame t (Figure 5.4a). Other modification proposals are in addition to frame t related to a trajectory ρ (we denote them collectively as $TrMod(\rho, t)$):

1. $Modify(\rho, t)$. Modify the shape of a trajectory ρ in frame t, i.e., use a different detection candidate (Figure 5.4b).

2. $\mathit{Prolong}(\rho, t)$. Prolong a trajectory ρ by one frame to the future or to the past. These proposals are considered only if t is either one frame before the initial frame of ρ or one frame after the final frame of ρ (Figure 5.4c).

3. $\mathit{Shorten}(\rho, t)$. Opposite of $\mathit{Prolong}$. These proposals are created only if t is either the first or the last frame of ρ (Figure 5.4c).

4. $\mathit{Append}(\rho, t)$. Append ρ to another trajectory. These proposals are created only if t is either the first or the last frame of ρ (Figure 5.4d).

5. $Cut(\rho, t)$. Cut ρ into two trajectories such that one of them ends in frame t (tail) and the other one starts in frame $t + 1$ (head) (Figure 5.4d).

6. $\mathit{Swap}(\rho, t)$. Cut ρ and another trajectory ρ' in frame t. Append the head of ρ to the tail of ρ' and, conversely, the head of ρ' to the tail of ρ (Figure 5.4e).

7. $\mathit{Split}(\rho, t)$. Split detection candidate representing frame t of ρ into two parts, use one of them to modify ρ and append the other one to another trajectory. These proposals are created only with detectors that allow to split detection candidates into two parts (Figure 5.4f).

8. $Merge(\rho, t)$. Opposite of Split – cut first/last frame of another trajectory and merge it with the detection candidate representing frame t of ρ (Figure 5.4f).

9. $\mathit{EstablishDivision}(\rho, t)$. Take two trajectories starting in frame $t + 1$ and make them offsprings of ρ. These proposals are created only if t is the last frame of ρ (Figure 5.4g).

10. $\mathit{CancelDivision}(\rho, t)$. Opposite of $\mathit{EstablishDivision}$ (Figure 5.4g).

Furthermore in every iteration the sampler considers a proposal that leaves the sample unchanged (we denote it as $NoChange$).

5.1.3 Transition Probabilities

In order to define the transition probabilities we will adopt the following notation. We denote the current object-level labeling by symbol X and use subscript i to denote a single variable and its label X_i. Formally, a modification proposal is an object-level labeling X' from a set of modification proposals (the set is in general different in every iteration). The transition probability $p_\theta(X'|X)$ is defined as a product of two terms. The first term is a feasibility function $\Gamma(X')$, which assigns 1 to feasible proposals and 0 to proposals that violate the feasibility constraints. The second term depends on a sum of potential functions ψ_f associated with individual variables. Every potential is a product of a log-density q_f of a feature f and a weight $w_{fX'_i}$:

$$\psi_f(X'_i, I; \theta) = w_{fX'_i} q_f(f(i, I), X'_i; \theta), \tag{5.1}$$

where I denotes the sequence of input images. Every variable might be associated with multiple features. For example detection variables might be associated with features that characterize their shape and/or brightness, features related to tracklets may quantify discrepancy between properties of the reference detection candidate (e.g. position, size, etc.) and the rest of the tracklet, etc. (see experimental section for features used in our experiments). We use Gaussian log-densities

$$q_f(f(i, I), X'_i; \theta) = -\log \sigma_{fX'_i} - \frac{\left(f(i, I) - \mu_{fX'_i}\right)^2}{2\sigma^2_{fX'_i}} \tag{5.2}$$

but the model can accommodate log-densities from different families as well. Parameter vector θ consists of the weights $w_{fX'_i}$ and parameters of the log-density, i.e., $\mu_{fX'_i}$ and $\sigma_{fX'_i}$. Transition probability of a modification proposal X' is then proportional to

$$p_\theta(X'|X, I) \propto \Gamma(X') \exp \sum_i \sum_{f \in \mathcal{F}_i} \psi_f(X'_i, I; \theta), \tag{5.3}$$

where $\mathcal{F}_i$ denotes the set of features associated with variable i.

5.1.4 MCMC Algorithm

The algorithm starts with the initial detection step. It iterates through the input sequence I and for each image generates an initial set of detection candidates using some initial (fixed) detector parameters $\theta_{\mathcal{D}_{init}}$. The algorithm then sets all $X_i = 0$ and repeatedly sweeps through the input sequence in order to generate a tracking sample. The pseudocode of one sweep is given in Algorithm 1. The algorithm iterates over all frames and in every frame t undergoes two phases. In the first one it goes over all trajectories ρ that are currently in the sample and modifies them using modification proposals from $TrMod(\rho, t)$. The second phase is devoted to adding new and removing solitary objects in frame t ($AddRemove(t)$ proposals). The second phase ends if the sample does not change for several iterations (5 in our experiments; denoted as $AddRemoveCondition(t)$ in the pseudocode).

```
for $t \leftarrow 1$ to $|I|$ do
    foreach Trajectory $\rho$ do
        $P \leftarrow \{NoChange\}$;
        foreach $Type(\rho, t) \in TrMod(\rho, t)$ do
            $P \leftarrow P \cup preselect(Type(\rho, t))$;
        end
        Sample $X' \in P$;
        $X \leftarrow X'$;
    end
    while AddRemoveCondition(t) do
        Sample $X' \in AddRemove(t) \cup \{NoChange\}$;
        $X \leftarrow X'$;
    end
end
```

Algorithm 1: One sweep of the MCMC algorithm.

During the first sweep the algorithm considers all feasible proposals. This behavior changes in the following sweeps: Before sampling a modification proposal in the trajectories-modification phase the algorithm first preselects for each type $Type(\rho, t) \in TrMod(\rho, t)$ a single proposal X'^{*} such that

$$X'^{*} = \underset{X' \in Type(\rho,t)}{\operatorname{argmax}} \; p_\theta(X'|X). \tag{5.4}$$

For certain types ($Modify(\rho, t)$, $Prolong(\rho, t)$ and $Split(\rho, t)$) the proposal X'^{*} is preselected not only from proposals available during the first sweep but the algorithm in addition optimizes the detector parameters $\theta_{\mathcal{D}}$ and generates new detection candidates in order to make new modification proposals. Since the properties of the new candidates depend on the current labeling X, this can be seen as a feedback. For example if the initial detection step fails to detect an object in certain frame, this mechanism allows the algorithm to make a new detection candidate which complies with the shape of that object in surrounding frames.

For many detectors it is not necessary to optimize the parameters $\theta_{\mathcal{D}}$ over the entire parameter space. For example, consider a detector composed of a neural network that classifies every pixel of an image as object or background by thresholding the output of the neuron in the final layer. Even a moderately large network contains millions of parameters and it would be infeasible to retrain it repeatedly every time the sampler preselects a modification proposal. Instead it may implement the feedback mechanism by changing the threshold only, which is tractable and still results in detection candidates not available during the first sweep.

5.1.5 Pixel-level Representation

As discussed in the previous section, the final tracking result is represented as a pixel-level labeling. Formally, it is a binary labeling S of pixels j of the input sequence I and edges jk connecting the pixels. Pixel labels S_j encode, whether a pixel j belongs to an object (label 1) or background (0) and edge labels S_{jk} encode, whether the connected pixels j and k belong to the same object (1) or not (0). Label 0 is also used for edges that connect a foreground pixel with a background pixel or two background pixels. Every pixel is connected with all other pixels in the same, previous and the next frame. This ensures that there is a path between every pair of pixels in the sequence and thus it is possible to determine their relationship. Edges between pixels in the same frame are called spatial, whereas edges connecting pixels in different frames are called temporal.

Every object-level variable corresponds to a small part of the pixel-level representation. Detection variables control labels of pixels that belong to the corresponding detection candidate and labels of spatial edges among these pixels. Tracklet variables control labels of temporal edges such that their both endpoints belong to detection candidates linked by the tracklet. And finally cell division variables control labels of temporal edges between the parent detection candidate and the offsprings.

Values of object-level variables X_i are translated to labeling S as follows: If the sampler sets value of an object-level variable to 1, the corresponding pixels and/or edges are labeled by 1. Setting label of an object-level variable to 0 does not necessarily result in setting labels of corresponding pixels/edges to 0. Instead, a pixel/edge is labeled by 0 if there is no object-level variable controlling its label (thus the mapping can be seen as logical or).

Since this mapping is deterministic, the probabilistic model defined in this section can be seen as a model of a subset of pixel-level labelings that correspond to some feasible object-level labeling. This observation is important for obtaining the final tracking result. As shown in the following section the inference task involves marginal expectations of labels of individual pixels and edges. These statistics are difficult to calculate exactly, but due to the deterministic mapping they can be estimated from the samples generated by the MCMC algorithm.

5.2 The Inference

In the Bayesian framework the final tracking result is a pixel-level labeling S^* that minimizes the expected loss (Bayes risk)

$$S^* = \operatorname*{argmin}_{\hat{S}} \mathbb{E}_{\pi_\theta} \left[l(S, \hat{S}) | I \right], \tag{5.5}$$

where π_θ denotes the probabilistic model. Since the space of possible decisions $\hat{S}$ is in our case the same as the space of events S, the loss function can be seen as a distance measure for pairs of tracking results. This observation motivates using pixel-level representation for the final result – as object-level representations of two tracking results are in general composed of different variables, there is no natural way for defining a distance function that would be efficient to calculate and in the same time had intuitive properties, i.e., it would assign high value to very different labelings and smaller value to similar ones. Our loss function is based on quadratic loss for binary labelings:

$$l(S,\hat{S}) = \sum_{j\in I}\left(S_j - \hat{S}_j\right)^2 + \frac{1}{|E_I|+1}\sum_{jk\in E_I}\left(S_{jk} - \hat{S}_{jk}\right)^2, \tag{5.6}$$

where E_I denotes the set of edges. The sum over edges is weighted by a constant, which ensures that the loss of the edge labeling is outweighed by the loss associated with a single pixel. This is motivated by the observation that a correct pixel labeling is a necessary condition for linking detected objects into trajectories and, therefore, it should receive higher priority.

The loss function (5.6) is defined for arbitrary labelings, even for those that cannot be interpreted as a tracking result. Consequently the optimal labeling S^* given by (5.5) is not guaranteed to be interpretable as a tracking result. To address this issue we alter the inference task (5.5) such that the optimal solution must in addition fulfill the following conditions:

1. For every pair of foreground pixels in the same frame and every pair of paths that are composed of spatial edges only and connect these pixels: all edges in these paths are either labeled by 1 or both paths contain at least one edge with label 0.

2. For every pair of foreground pixels and every pair of paths that are composed of temporal edges only and connect these pixels: all edges in these paths are either labeled by 1 or both paths contain at least one edge with label 0.

A labeling that fulfills these conditions is called "physically interpretable". If we substitute the loss function (5.6) into (5.5) and discard terms that cannot influence the optimal solution, we end up with the following optimization task:

$$\begin{aligned} S^* = \operatorname*{argmin}_{\hat{S}} & \sum_{j\in I}\hat{S}_j\left(1 - 2\mathbb{E}_{\pi_\theta}\left[S_j|I\right]\right) + \\ & \frac{1}{|E_I|+1}\sum_{jk\in E_I}\hat{S}_{jk}\left(1 - 2\mathbb{E}_{\pi_\theta}\left[S_{jk}|I\right]\right) \\ \text{s.t.} \quad & S^* \text{ is physically interpretable.} \end{aligned} \tag{5.7}$$

Expectations $\mathbb{E}_{\pi_\theta}\left[S_j|I\right]$ and $\mathbb{E}_{\pi_\theta}\left[S_{jk}|I\right]$ can be estimated from samples generated by the MCMC algorithm. Due to the small weight imposed on the edge-related term of (5.7) the optimization can be done separately for pixels and edges. Because no labeling of pixels can violate the interpretability, the first part can be solved by choosing the label of each pixel independently:

$$S_j^* = \begin{cases} 0 & \mathbb{E}_{\pi_\theta}\left[S_j|I\right] < 0.5 \\ 1 & \mathbb{E}_{\pi_\theta}\left[S_j|I\right] \geq 0.5. \end{cases} \tag{5.8}$$

The optimization of the edge-related term of (5.7) is however a difficult energy minimization problem. Physically interpretable binary labeling of edges is equivalent to a multiway cut, i.e., separating the vertices of the graph into several connected components. Finding a minimum cost multiway cut is in general NP hard [43] and in our case the problem is even more complicated, because the number of connected components is not given beforehand. Therefore to obtain a solution we minimize the edge-related term by a greedy algorithm.

The algorithm starts with all edges labeled by 0, which means that every pixel is considered a separate connected component. The connected components are then repeatedly merged (by changing labels of the edges between them to 1): in every iteration the algorithm selects a pair of connected components with largest possible decrease of risk (5.7). The algorithm stops when the risk cannot be further decreased by merging more components.

As expectations $\mathbb{E}_{\pi_\theta}\left[S_{jk}|I\right]$ are estimated from finite number of samples it often happens in practice that there are large groups of pixels that always belong to the same object. Using these groups as initial connected components speeds up the optimization considerably and we found empirically that it leads to a very similar solution as starting from individual pixels.

5.3 Parameter Learning

Our method makes use of two types of parameters: the initial detector parameters $\theta_{\mathcal{D}_{init}}$ and the model parameters θ. We assume that $\theta_{\mathcal{D}_{init}}$ are given beforehand and concentrate on the parameters of the model.

The parameters θ are learned from training data consisting of pairs of input sequences and ground truth annotations given as pixel-level labelings. A common estimator of parameters of probabilistic models is the maximum likelihood (ML) estimator, which maximizes the probability of the training data given the model. However, in our case the ML estimator suffers from a fundamental flaw – due to limitations of the detector the MCMC algorithm may not be able to generate the ground truth labeling, in which case it would have zero probability regardless of parameters of the model. Furthermore calculating probabilities of possible labelings is intractable, because it involves summation over vast number of runs of the MCMC algorithm.

To tackle these issues we use an alternative learning objective. For each training sample we define an "ideal" run of the MCMC algorithm, in which the algorithm quickly generates a labeling similar to the ground truth annotation. The parameters θ are then learned such that these ideal runs have high probability. The ideal run associated with annotation S is defined in terms of modification proposals the algorithm would select in each iteration. Suppose that after a iterations it generated a labeling $S^{(a)}$ and considers modification proposals $S'^{(a)}$ from a set of available proposals $P^{(a)}$. The selected proposal $S^{(a+1)}$ is such that it maximizes the decrease of loss

$$S^{(a+1)} = \operatorname*{argmax}_{S'^{(a)} \in P^{(a)}} \Delta_S(S'^{(a)}, S^{(a)}) = l(S^{(a)}, S) - l(S'^{(a)}, S). \tag{5.9}$$

Note that although the algorithm uses the object-level representation, for the sake of brevity we refer to the corresponding pixel-level labelings. The ideal run should incorporate at least two sampling sweeps – the initial one and one more with the feedback mechanism. Since the ideal run can be performed easily for arbitrary S, for convenience we consider the resulting sequence $S^{(1)}..S^{(n)}$ part of the training data (note that n is in general different for each S).

A simple option, how to learn the parameters θ, such that the ideal runs have high probability, is to define a learning objective, which maximizes the probability that in every state $S^{(a)}$ of an ideal run the sampler selects a modification proposal $S'^{(a)}$, which is identical with the next state $S^{(a+1)}$ of that ideal run. This approach would however suffer from two major issues. In every step the sampler can typically choose from multiple modification proposals and considering only one of them as "correct" would in fact lead to imbalanced training data. Furthermore in some cases many of the alternative proposals are similar (in terms of loss (5.6)) to the "correct" one and learning the parameters such that they are not likely to be selected could negatively influence the probability of the "correct" proposal as well and lead to a poorly trained model.

To overcome these issues we consider the expected decrease of loss associated with the a-th step of an ideal run $S^{(1)}..S^{(n)}$:

$$g_\theta(S^{(a)}) = \sum_{S'^{(a)} \in P^{(a)}} \Delta_S(S'^{(a)}, S^{(a)}) p_\theta(S'^{(a)} | S^{(a)}, I). \tag{5.10}$$

Maximizing $g_\theta(S^{(a)})$ corresponds to maximizing the probability, that the sampler selects a modification proposal, which is identical or *similar* to the next state of the ideal run. This is a crucial difference from maximizing $p_\theta(S^{(a+1)}|S^{(a)}, I)$ directly, because the modification proposals are no longer splited to correct and incorrect. The learning objective is then defined as a sum of expected decreases of loss over all iterations of all ideal runs in the training data:

$$\theta^* = \operatorname*{argmax}_\theta \sum_{(I,S,S^{(1)}..S^{(n)}) \in \mathcal{T}} \sum_{a=1}^{n} g_\theta(S^{(a)}) \tag{5.11}$$

where $\mathcal{T}$ denotes the set of training data. This objective is tractable and differentiable and we solve it by gradient descent method with backtracking line search.

5.4 Experiments

We test our method on two different datasets: time-lapse phase contrast microscopy image sequences of endothelial cells and a subset of the ISBI Cell Tracking Challenge. Due to their different appearance, we describe the detectors, used features and results for each dataset separately.

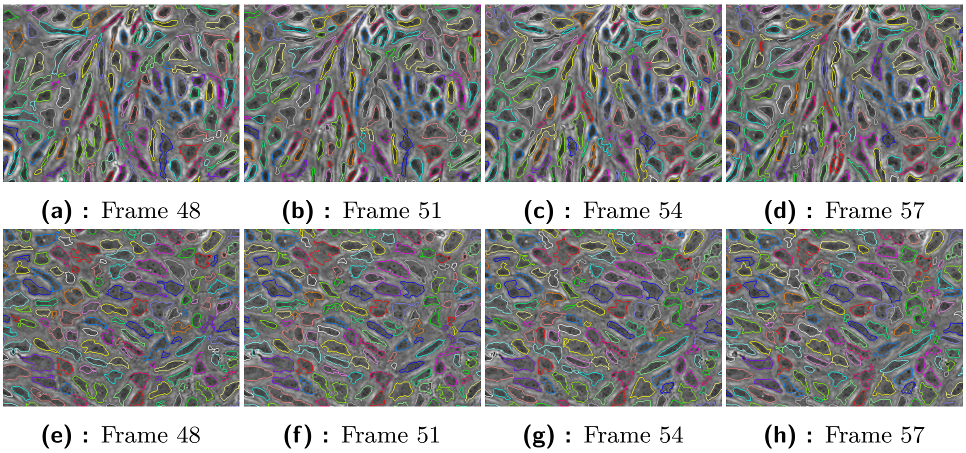


Figure 5.5: Visualization of tracking results for endothelial cells treated with dimethyl sulfoxide (a) - (d) and EHT1864 inhibitor (e) - (h) (experiment E3). For the sake of exposition we cropped the input images.

5.4.1 Endothelial Cells

The experiments with the endothelial dataset were conducted in cooperation with Jiahui Cao, Jochen Seebach and Hans Schnittler from Institute of Anatomy and Vascular Biology, Faculty of Medicine, Westfälische Wilhelms University of Münster [29]. The data consists of 37 unannotated time-lapse phase contrast microscopy sequences of confluent human umbilical vein endothelial cells isolated from umbilical cord veins of different donors (Figure 5.5). The cells were placed on a gelatin-coated 12-well plate and automatically imaged with an Axio observer Z1 (Carl Zeiss, Oberkochen, Germany) supplied with humidity control module and 5% CO_2 using 10× and 20× plan objectives. The sequences come from six different experiments (E1 - E6), each verifying the effect of a certain biochemical treatment to cell elongation and migration ability. In every experiment we recorded several videos of proliferating cells exposed to the treatment (Group 1) and several more serving as a control (Group 2). The resolution of the resulting videos is 1388×1040 px, their length varies from 46 to 157 frames and they contain several hundred to several thousands cells. The treatments and detailed properties of the recorded sequences are summarized in appendix A. The dataset can be obtained from Hans Schnittler (hans.schnittler@uni-muenster.de) upon request.

Detector

The input images are first smoothed by a small Gaussian filter: 5×5 pixels for sequences observed with 10× objective ($\sigma = 4.5$) and 11×11 pixels for sequences observed with 20× objective ($\sigma = 7.5$). We make use of observation that the endothelial cells appear as dark objects surrounded by brighter areas. The detector is based on the watershed transform. Starting from a given seed (local minimum of the image brightness) it floods the basin representing a cell and in the process keeps track of the brightest pixel flooded so far. The flooding continues for a predefined number of pixels and a new detection candidate is created every time the brightest pixel changes at least λ_{ws} times in a row. Consequently there are typically several candidates created from one seed. The initial detection step is performed for all local minima of the image brightness. The threshold λ_{ws} is learned by exhaustive search using several annotated cells in a few input images such that all annotated cells are found correctly and the total number of generated candidates is minimized.

Detector Feedback

A value of λ_{ws} optimal for one image may not be the best for another one and as a result, the initial detection step may not find shapes of some cells accurately or even miss them completely. Decreasing λ_{ws} such that the detector finds all cells accurately in all frames may significantly increase the number of initial detection candidates and consequently the computational cost and therefore we rather tackle this issue by the feedback mechanism. When constructing a shape modification or trajectory prolongation proposal after the first sweep the detector is run again from several nearby local minima of the brightness with $\lambda_{ws} = 1$. Using this setting a new detection candidate is created every time the value of the brightest flooded pixel increases which allows the sampler to determine cell shapes more accurately without significant increase of computational cost. The splitting and merging proposals are not used for endothelial cells.

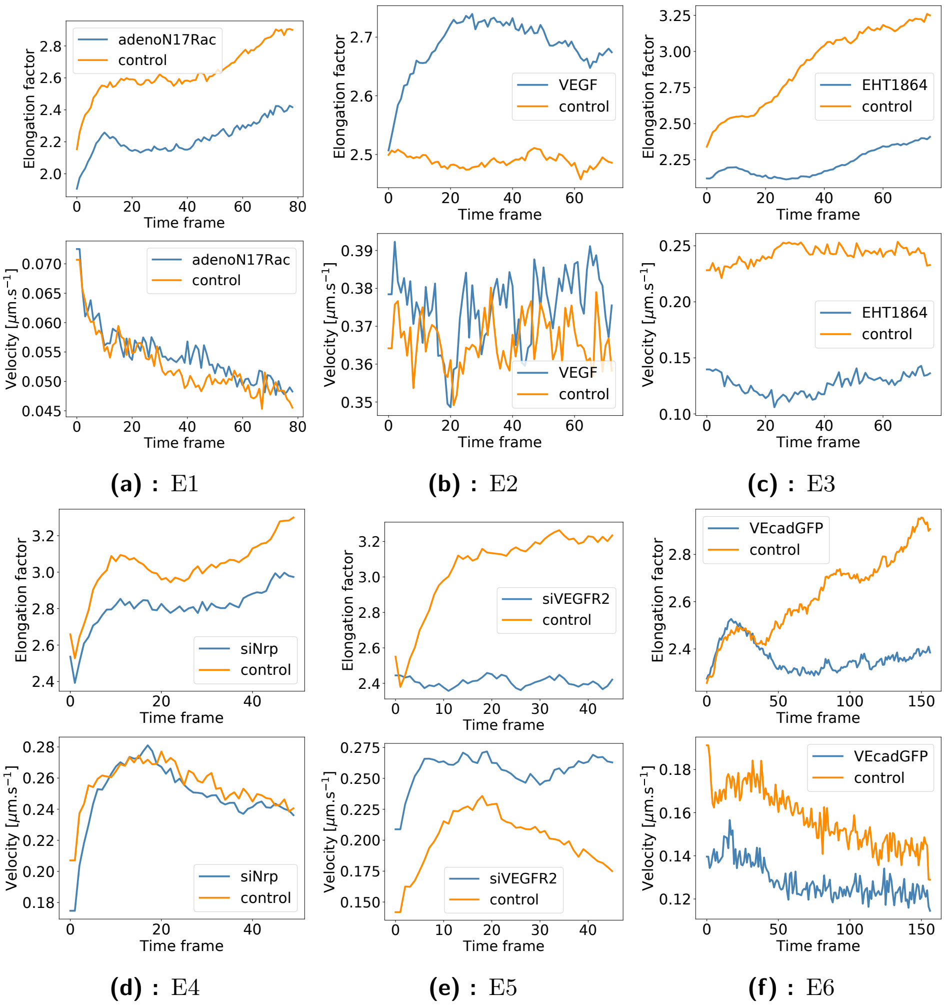
Features

We used the following features f associated with the detection variables:

1. **Ellipticity:** A real number between 0 and 1 characterizing, how much the detection candidate resembles an ellipse (1 means perfect ellipse) (see [158] for exact definition).
2. **Boundary stability:** A real number between 0 and 1 indicating, whether the created detection candidate would be inflated in all directions by further flooding or whether the newly flooded pixels would belong to a different drainage basin. When a detection candidate is created, the flooding continues for additional n_b steps, where n_b is the length of the exterior boundary, i.e. pixels, that are not part of the candidate but at least one of their neighboring pixels belongs to the candidate. When this additional flooding finishes some pixels from the original exterior boundary will be flooded (we denote their number by n_f). The value of the feature is then $1 - \frac{n_f}{n_b}$.

Features associated with the tracklets:

3. **Motion position:** Euclidean distance between the centroid of the reference detection candidate and its position linearly interpolated from the neighboring frames.



(a) : E1

(b) : E2

(c) : E3

(d) : E4

(e) : E5

(f) : E6

Figure 5.6: Plots of elongation and migration ability of endothelial cells exposed to various biochemical treatments.

4. **Motion shape:** Discrepancy between shape of the reference detection candidate and a shape interpolated from the remaining detection candidates of the tracklet. The value of the feature is calculated in four steps:

(4.1) The detection variables linked by the tracklet are transformed into sets of pixels, each composed of pixels that belong to the corresponding detection candidate. The set corresponding to the reference candidate is denoted as D^{Ref} and the other sets as $D^1..D^{l-1}$, where l is the length of the tracklet. Note that for the purpose of calculating this feature all detection candidates are considered to be in a single frame.

(4.2) Sets $D^1..D^{l-1}$ are shifted such that their centroids become equal to the centroid of D^{Ref}.

(4.3) Sets $D^1..D^{l-1}$ are combined into a single fuzzy set D^{Int} such that its characteristic function $\mu_{D^{Int}}(j)$ for a pixel j is given as a number of sets from $D^1..D^{l-1}$ that contain j divided by $l-1$.

(4.4) The value of the feature is the Jaccard index $\frac{|D^{Ref} \cap D^{Int}|}{|D^{Ref} \cup D^{Int}|}$, where $\cap$ is pixel-wise minimum and $\cup$ is pixel-wise maximum.

Both motion features involve two frames in the future and two in the past.

Due to sparsity of cell division events in the data we did not use the cell division variables in the model.

Results

We used our method to determine the effects of various biochemical treatments to elongation and migration ability of endothelial cells. For every sequence we calculated frame-wise averages of cell velocities and their elongation factors defined as $e = \sqrt{\frac{|\lambda_1|}{|\lambda_2|}}$, where λ_1 and λ_2 are the eigenvalues of estimated covariance matrix of the cell's shape D

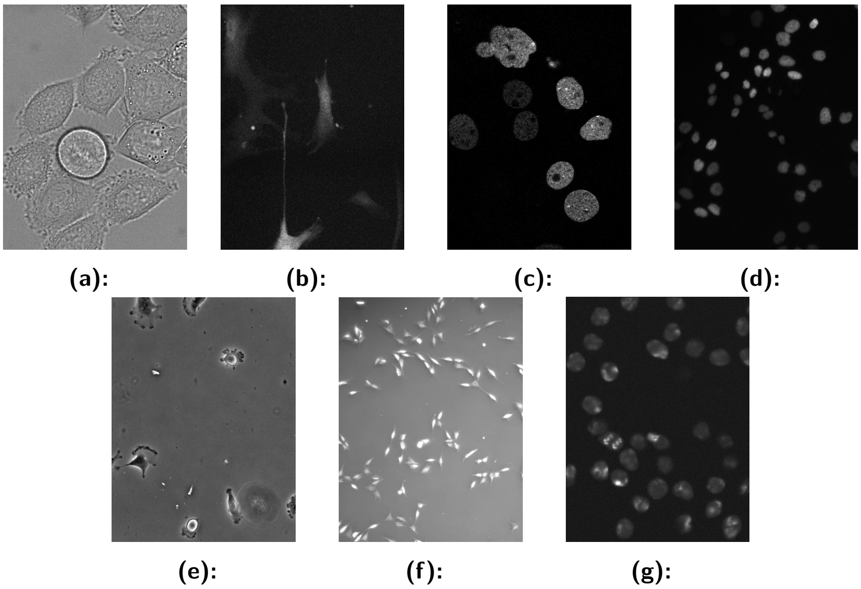
$$\frac{1}{|D|} \sum_{j \in D} (j - c(D)) (j - c(D))^\top \tag{5.12}$$

and $|\lambda_1| \geq |\lambda_2|$ (note that we treat D as a set of pixels with centroid $c(D)$). The final analysis was based on frame averages of velocities and elongations of each group.

The results along with visualization of estimated cell shapes and their trajectories are shown in Figure 5.6 and Figure 5.5, respectively. In experiments E1, E3, E4, E5 and E6 the selected treatment blocked the ability of cells to elongate, although in E4 and E6 the effect did not take place immediately from the beginning of the sequence. In experiment E2 the treatment encouraged the cells to elongate. In experiments E3 and E6 the treatment blocked the ability of cells to move, whereas in E5 the cells transfected with siVEGFR2 were able to move faster than cells in the control group. The treatment in experiments E1, E2 and E4 had no measurable effect on cells migration ability. For more details see [29].

5.4.2 ISBI Cell Tracking Challenge

As the endothelial dataset lacks ground truth annotations and therefore does not allow for quantitative evaluation, we evaluate our method on data from the ISBI Cell Tracking Challenge [125]. The aim of the challenge is to compare various tracking methods on diverse datasets of time-lapse videos of moving cells. We test our method on 2D datasets (the same dimensionality as the endothelial dataset) from the challenge: HeLa cells on a flat glass (DIC-C2DH-HeLa), rat mesenchymal stem cells on a flat polyacrylamide substrate (Fluo-C2DL-MSC), GFP-GOWT1 mouse stem cells



(a): (b): (c): (d):

(e): (f): (g):

Figure 5.7: Selected datasets from the ISBI Cell Tracking Challenge. DIC-C2DH-HeLa (a). Fluo-C2DL-MSC (b). Fluo-N2DH-GOWT1 (c). Fluo-N2DL-HeLa (d). PhC-C2DH-U373 (e). PhC-C2DL-PSC (f). Fluo-N2DH-SIM+ (g). For the sake of exposition we cropped the input images and increased their contrast.

(Fluo-N2DH-GOWT1), HeLa cells stably expressing H2b-GFP (Fluo-N2DL-HeLa), glioblastoma-astrocytoma U373 cells on a polyacrylimide substrate (PhC-C2DH-U373), pancreatic stem cells on a polystyrene substrate (PhC-C2DL-PSC) and simulated nuclei of HL60 cells stained with Hoescht (Fluo-N2DH-SIM+) (Figure 5.7).

Due to their diverse appearance it is very challenging for a single detector to work with all datasets: the dimensions of objects vary from only a few pixels to hundreds of pixels, there is high diversity in brightness (sometimes even within one sequence) and in some cases the training sequences have different photometric properties than the testing videos. Every dataset contains two annotated training sequences and two unannotated testing sequences. With the exception of Fluo-N2DH-SIM+ the training videos are only partially annotated which further adds to the challenges the tracker has to contend with. Tracking results for testing sequences can be submitted and evaluated using an online system, which guarantees an unbiased method comparison.

Detector

The backbone of the detector is a multi-layer perceptron neural network with four hidden layers (each 64 neurons with ReLu activation) and two output neurons (sigmoid activation), which for each pixel j predicts the probability of being part of a cell (segmentation) and truncated distance B_j^{NN} to the nearest cell boundary. Since the sigmoid can only attain values between 0 and 1, the output of the boundary neuron is multiplied by the truncation threshold: $B_j^{NN} \leftarrow \lambda_{dist} * B_j^{NN}$. The input layer is a square window of size

	λ_{rf}	λ_{dist}	λ_{seg}
DIC-C2DH-HeLa	20	19	0.8
Fluo-C2DL-MSC	13	5	0.5
Fluo-N2DH-GOWT1	13	5	0.4
Fluo-N2DL-HeLa	13	12	0.05
PhC-C2DH-U373	7	6	0.4
PhC-C2DL-PSC	15	14	0.4
Fluo-N2DH-SIM+	7	1	0.5

Table 5.1: Hyperparameters of the detector used for the ISBI Cell Tracking Challenge

$\lambda_{rf} \times \lambda_{rf}$. Values of λ hyperparameters are summarized in Table 5.1.

The output of the segmentation neuron is thresholded at λ_{seg} and every foreground blob (a connected component of foreground pixels) is partitioned into one or several detection candidates. The partitioning algorithm has three steps: first it finds clustering seeds, then it precalculates distances between seeds and other pixels and finally it selects a subset of seeds and assigns each pixel to the nearest selected seed. The seeds are local maxima of boundary distance transform of the blob, which is obtained by calculating L^1 distance of each pixel to the nearest boundary. In the second step the distance between two pixels is defined as the cost of the shortest path between these pixels, where the cost of stepping on a pixel j is $\lambda_{dist} - B_j^{NN}$. This definition helps to create clusters that closely follow cell boundaries predicted by the neural network. The subset of seeds is selected as follows: Initially the blob is partitioned such that the discrepancy between predicted boundary distance transform B^{NN} and the actual B induced by the created clusters

$$\sum_{j \in \text{blob}} \left(B_j^{NN} - B_j \right)^2 \tag{5.13}$$

is minimized. After initialization (first sweep) the tracker is allowed to reconsider the initial partitioning and select a different one, which is more probable with respect to the tracking model (see subsection 5.4.2).

The neural network was trained using fully annotated images (every training sequence has at least two of them). The hyperparameters λ were selected exhaustively such that the resulting neural network was able to achieve the smallest possible validation error.

Detector Feedback

The initial detection step sometimes mistakenly merges one cell with another or splits a cell into several parts, but detects it correctly in a neighboring frame. This is handled by a split/merge proposal, which revisits the corresponding blob into a different number of detection candidates. In our implementation we only consider split/merge proposals, which change the number of cells by one.

	no feedback		feedback	
	TRA	SEG	TRA	SEG
DIC-C2DH-HeLa	0.618	0.522	**0.818**	**0.720**
Fluo-C2DL-MSC	**0.850**	**0.615**	**0.850**	**0.615**
Fluo-N2DH-GOWT1	0.976	0.874	**0.981**	**0.933**
Fluo-N2DL-HeLa	0.957	0.754	**0.980**	**0.815**
PhC-C2DH-U373	0.934	0.884	**0.960**	**0.909**
PhC-C2DL-PSC	0.791	0.676	**0.918**	**0.719**
Fluo-N2DH-SIM+	0.916	0.745	**0.954**	**0.782**

Table 5.2: Evaluation of the feedback mechanism.

Features

We used the following features f associated with the detection variables:

1. **Ellipticity:** Same definition as in 5.4.1.

Features associated with the tracklets:

2. **Motion position:** Same definition as in 5.4.1.

3. **Motion size:** Ratio of size of the reference detection candidate and the size linearly interpolated from the neighboring frames. Both motion features involve one frame in the future and one in the past.

Features associated with the cell division variables:

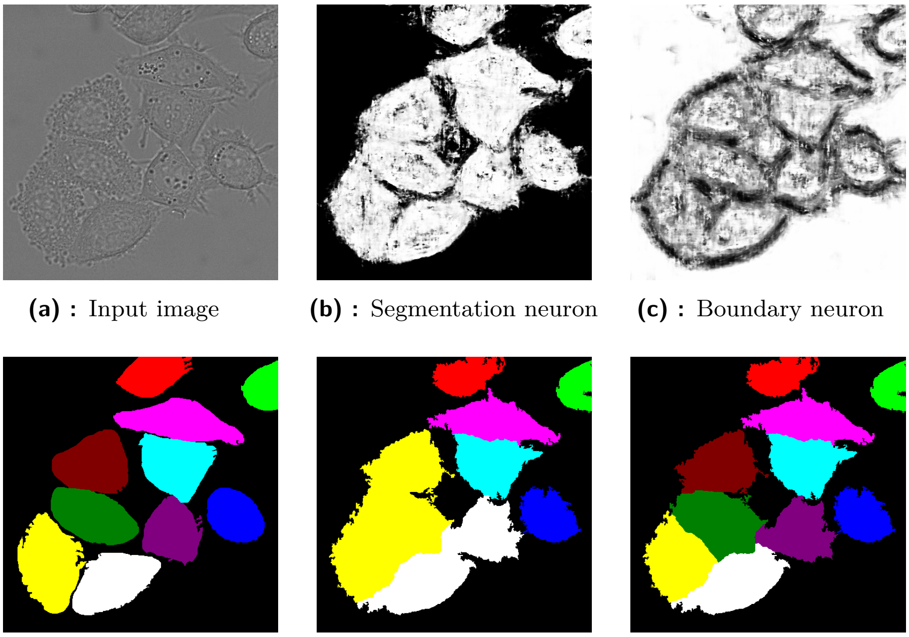
4. **Division-brightness:** Ratio of the average brightness of the candidates involved directly in the cell division and the average brightness of candidates in the parent and offspring trajectories in the neighboring frames.

5. **Division-motion:** Overlap of the offspring candidate with the parent divided by size of the offspring. The value of the feature is average over both offsprings.

Evaluation of the Feedback Mechanism

Despite its universality the performance of the detector proved to be unsatisfactory for some datasets and the quality of the detection candidates (and consequently the tracking results) could be improved only by the detector feedback mechanism. To quantify the improvement we run our method on selected datasets without the feedback (i.e., the algorithm was not allowed to consider split/merge modification proposals) and compared the output with results obtained using the feedback mechanism. In both scenarios we applied the method to the training (annotated) sequences, each time one being the input of the tracker and the other one used for parameter learning.

To make sure that the improvement can be attributed to the feedback mechanism alone and it is not affected by ad hoc image processing techniques,



(a) : Input image

(b) : Segmentation neuron

(c) : Boundary neuron

(d) : Ground truth segmentation

(e) : Initial detection candidates

(f) : Corrected detection candidates

Figure 5.8: Output of the detector using an input image from the DIC-C2DH-HeLa dataset.

we did not preprocess the videos in any way and the only postprocessing involved filling holes in tracked cells. The quality of results was measured by metrics from the challenge: the tracking precision TRA (precision of the tracking graph) and the segmentation accuracy SEG (they can be calculated using a software provided with the challenge data). They both fall into the $[0, 1]$ interval with higher values corresponding to better performance. For their detailed description see the challenge webpage[1]. For every dataset we report average values of TRA and SEG metrics.

The results are summarized in Table 5.2. The feedback mechanism is especially effective in datasets, which contain closely packed cells with poorly visible boundaries: DIC-C2DH-HeLa and PhC-C2DL-PSC. Its importance is lower for sparser datasets: Fluo-N2DH-GOWT1, Fluo-N2DL-HeLa, PhC-C2DH-U373 and Fluo-N2DH-SIM+. The dataset Fluo-C2DL-MSC is an extreme case with no visible cell-to-cell interactions and as a result the feedback mechanism implemented purely by split/merge proposals was not able to improve the performance of the method.

Sample output of the detector and the effect of the feedback mechanism is visualized in Figure 5.8. The cells in the input image from the DIC-C2DH-HeLa dataset (Figure 5.8a) are closely packed and although the detector is able to accurately classify pixels as background and foreground (Figure 5.8b), it struggles to predict for each pixel its distance to a nearest boundary between two cells (Figure 5.8c). As a result the detector mistakenly merges

[1] http://www.celltrackingchallenge.net/

	#1	#2	#3	Ours	Rank/#Methods
DIC-C2DH-HeLa	0.881_a	0.797_b	0.752_c	0.780	3/5
Fluo-C2DL-MSC	0.873_d	0.763_b	0.691_a	0.737	3/14
Fluo-N2DH-SIM+	0.975_a	0.957_b	0.948_j	0.935	4/15
Fluo-N2DL-HeLa	0.991_b	0.986_e	0.982_f	0.987	2/15
PhC-C2DH-U373	0.981_a	0.977_b	0.965_h	0.957	4/6
PhC-C2DL-PSC	0.943_f	0.942_b	0.898_i	0.862	6/9

(a) : TRA

	#1	#2	#3	Ours	Rank/#Methods
DIC-C2DH-HeLa	0.776_a	0.460_b	0.293_c	0.464	2/5
Fluo-C2DL-MSC	0.645_d	0.590_b	0.582_a	0.579	4/14
Fluo-N2DH-SIM+	0.791_b	0.781_a	0.770_j	0.694	7/15
Fluo-N2DL-HeLa	0.903_a	0.893_b	0.863_g	0.869	3/15
PhC-C2DH-U373	0.920_a	0.826_h	0.795_b	0.833	2/6
PhC-C2DL-PSC	0.665_f	0.602_b	0.572_i	0.622	2/9

(b) : SEG

	#1	#2	#3	Ours	Rank/#Methods
DIC-C2DH-HeLa	0.828_a	0.629_b	0.523_c	0.622	3/5
Fluo-C2DL-MSC	0.759_d	0.676_b	0.636_a	0.658	3/14
Fluo-N2DH-SIM+	0.878_a	0.874_b	0.859_j	0.814	7/15
Fluo-N2DL-HeLa	0.942_b	0.940_a	0.901_e	0.928	3/15
PhC-C2DH-U373	0.951_a	0.896_h	0.886_b	0.895	3/6
PhC-C2DL-PSC	0.804_f	0.772_b	0.735_i	0.742	3/9

(c) : OP

Table 5.3: Leaderboard of the ISBI Cell Tracking Challenge in the time of submitting our results (for each dataset the top three results were achieved by different methods). Tracking precision TRA (a), segmentation accuracy SEG (b) and the overall performance OP=(TRA+SEG)/2 (c). Top three results were achieved by these participants: a: FR-Ro-GE, b: KTH-SE, c: IMCB-SG, d: BGU-IL, e: HD-Har-GE, f: HD-Hau-GE, g: UZH-CH, h: FR-Be-GE, i: UP-PT, j: PAST-FR. Our method consistently scored among the best three contenders.

several cells as shown in Figure 5.8e. However, as the corresponding cells are detected more accurately in the neighboring frames, their shapes propagate to the visualized frame and the boundaries are corrected (Figure 5.8f). In this case the detection errors were corrected after the third sweep of the MCMC algorithm.

Online Evaluation

We submitted the results for six datasets using the same settings of the method (hyperparameters, pre and postprocessing) as in the previous experiment. For every dataset we used a single model trained on both training sequences. Due to different photometric properties of the training and the testing sequences of Fluo-N2DH-GOWT1 the detector was not able to achieve

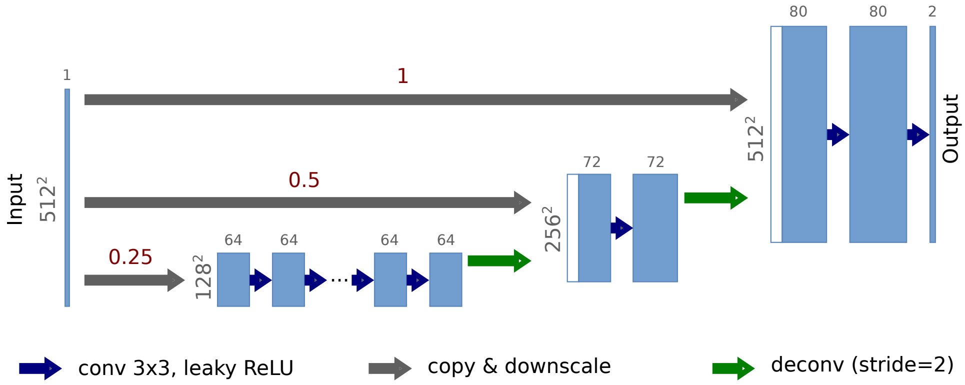


Figure 5.9: Multiresolution convolutional neural network used as a detector in the second submission (example for 512 × 512 px images and three segments). Each blue box represents a single convolutional layer preceded by a padding layer (reflection of the boundary pixels), which is used to preserve the dimensions of the input tensor. The numbers above the gray arrows indicate the corresponding resolution (e.g. 0.25 means downscaling by factor four). The white boxes represent copied input image. The numbers above the blue boxes indicate the number of channels of the convolutional layer.

acceptable performance (even with the feedback) without extensive data preprocessing and consequently the results for this dataset were not included in the submission.

The results are summarized in Table 5.3. Besides TRA and SEG metrics we also include the overall performance score defined as (TRA+SEG)/2. Despite its universal and simplistic detector our method consistently belonged among the best three contenders (for each dataset the top three results are achieved by different methods).

5.4.3 ISBI Cell Tracking Challenge – Second Submission

To improve the results achieved in the first submission we enhanced the detection neural network and resubmitted the updated results. In this subsection we describe the differences from the first submission and discuss the improvement in the performance.

Detector

Unlike in the first submission, where the neural network was used as a pixels classifier, the detector is a multiresolution convolutional neural network (CNN) trained in end-to-end fashion. The network is composed of three segments, each being a CNN with 3 × 3 convolution filters and leaky ReLU activations (Figure 5.9). The first segment is a CNN with 16 layers and the input to the first layer is an input image downscaled by factor 4. The segment is followed by a deconvolution layer, which upscales the output of the last layer by factor two. The upscaled output is then concatenated with the input image downscaled to the corresponding resolution and used as the input to

	#1	#2	#3	Ours	Rank/#Methods
DIC-C2DH-HeLa	0.915_k	0.881_a	0.797_b	0.898	2/10
Fluo-C2DL-MSC	0.873_d	0.765_k	0.763_b	0.720	5/17
Fluo-N2DH-GOWT1	0.976_b	0.947_k	0.932_d	0.904	9/18
Fluo-N2DH-SIM+	0.975_a	0.966_d	0.957_b	0.955	5/20
Fluo-N2DL-HeLa	0.991_b	0.987_o	0.986_e	0.988	2/20
PhC-C2DH-U373	0.983_k	0.981_a	0.977_b	0.974	4/10
PhC-C2DL-PSC	0.943_f	0.942_b	0.934_d	0.925	4/13

(a) : TRA

	#1	#2	#3	Ours	Rank/#Methods
DIC-C2DH-HeLa	0.814_k	0.776_a	0.511_d	0.792	2/10
Fluo-C2DL-MSC	0.645_d	0.617_l	0.590_b	0.579	6/17
Fluo-N2DH-GOWT1	0.927_b	0.893_m	0.887_n	0.894	2/18
Fluo-N2DH-SIM+	0.802_d	0.791_b	0.781_a	0.807	1/20
Fluo-N2DL-HeLa	0.903_a	0.902_l	0.893_b	0.900	3/20
PhC-C2DH-U373	0.924_l	0.920_a	0.846_d	0.922	2/10
PhC-C2DL-PSC	0.665_f	0.633_l	0.626_d	0.682	1/13

(b) : SEG

	#1	#2	#3	Ours	Rank/#Methods
DIC-C2DH-HeLa	0.864_k	0.828_a	0.629_b	0.845	2/10
Fluo-C2DL-MSC	0.759_d	0.676_b	0.658_o	0.649	4/17
Fluo-N2DH-GOWT1	0.951_b	0.914_k	0.902_m	0.899	5/18
Fluo-N2DH-SIM+	0.882_d	0.878_a	0.874_b	0.881	2/20
Fluo-N2DL-HeLa	0.942_b	0.940_l	0.940_a	0.944	1/20
PhC-C2DH-U373	0.951_a	0.936_l	0.902_d	0.948	2/10
PhC-C2DL-PSC	0.804_f	0.780_d	0.772_b	0.804	2/13

(c) : OP

Table 5.4: Leaderboard of the ISBI Cell Tracking Challenge in the time of the second submission (for each dataset the top three results were achieved by different methods). Tracking precision TRA (a), segmentation accuracy SEG (b) and the overall performance OP=(TRA+SEG)/2 (c). Top three results were achieved by these participants: a: FR-Ro-GE, b: KTH-SE, d: BGU-IL, e: HD-Har-GE, f: HD-Hau-GE, k: TUG-AT, l: FR-Fa-GE, m: LEID-NL, n: CUNI-CZ, o: CVUT-CZ (our original submission). In this summary we consider our original submission as an independent method. Note that as the challenge is open for new online submissions, the standings may change in the future.

the second segment (CNN with 2 layers). The output of the second segment is again upscaled by a deconvolution layer, concatenated with the (original resolution) input image and used as the input to the final segment (CNN with 2 layers). The last layer predicts for each pixel j the probability of being part of a cell (the corresponding neurons have sigmoid activations) and truncated distance B_j^{NN} to the nearest cell boundary (ReLU activations). Due to the ReLU activations of the distance related neurons the network is able to predict B_j^{NN} directly and there is no need to explicitly multiply the output by a truncation threshold. The output of the network is used to create detection candidates in the same way as in the first submission.

The neural network was trained using fully annotated images. To compensate for small number of annotated images we used data augmentation extensively. The images were randomly flipped (horizontally or vertically) or rotated by 90°, 180° or 270° and transformed by an elastic transform [178]. Furthermore, in the images from the Fluo-N2DH-GOWT1 dataset we added a small random number to the intensity of foreground pixels in order to compensate for the different photometric properties of the training and testing sequences.

Online Evaluation

The results of the second submission are summarized in Table 5.4. With the exception of the Fluo-C2DL-MSC dataset the new detector helped to achieve better performance in all three objectives. Although the number of competing methods increased due to new submissions, the improved performance resulted in the same or even better absolute position of our method in the challenge rankings for majority of the datasets. Furthermore, the new detector and the extensive use of data augmentation also allowed to submit results for the Fluo-N2DH-GOWT1 dataset which was missing in the original submission. Our method is also absolute winner in the overall performance (OP) objective for the Fluo-N2DL-HeLa dataset as well as in the segmentation (SEG) objective for the Fluo-N2DH-SIM+ and PhC-C2DL-PSC datasets. Note that as the challenge is open for new online submissions, the standings may change in the future.

5.5 Summary

We proposed a joint model for segmentation, detection and tracking of biological cells. The model is defined implicitly in terms of a Markov chain Monte Carlo algorithm, which gives it great flexibility without making the inference intractable. It contains a temporal feedback that allows to dynamically create new detection candidates based on hints from surrounding frames and in this way to overcome detection errors that would otherwise propagate into the data association module. This reduces the need for a complex and difficult to design detection module and helps the method to achieve competitive results using a simple general purpose detector. The parameters of the model are learned using an objective based on empirical risk minimization. We evaluated our method on selected datasets from the ISBI Cell Tracking Challenge, where it consistently belonged among the best three methods and used it to conduct large-scale experiments with confluent cultures of endothelial cells that investigated the effect of various biochemical treatments to cell elongation and migration ability.

Chapter 6
Conclusions

In this thesis we focused on joint segmentation, detection and tracking of biological objects. We presented three different approaches that tackle this problem as a whole or are related to particular subproblems.

First, we presented a novel method for tracking dimensionless particles. It is based on a probabilistic graphical model for sets of moving objects, each represented by its trajectory, and it allows to model mutual interactions among the objects. The tracker does not assume, that a fixed set of detection candidates was generated beforehand but instead uses a map of detection scores to infere the number and positions of objects. Although this cannot be seen as fully joint model of detection and tracking, it nevertheless allows to overcome some detection errors. This observation is supported by experimental evaluation on a dataset from the 2012 ISBI Particle Tracking Challenge, where our approach proved to be competitive with state-of-the-art methods.

Our second contribution is a probabilistic shape model for cell segmentation. The model is a Markov Random Field on bipartite graph, such that variables of one layer represent a segmentation of the corresponding pixels and the variables in the second layer serve as a regularizer for the segmentation within their receptive fields. In addition, we proposed a novel algorithm for unsupervised parameter learning for this class of models. The algorithm is a modified EM algorithm, which replaces the likelihood estimator in the M step by the pseudolikelihood. The learning algorithm was compared with the Persistent Contrastive Divergence and it was shown to be more stable and to converge faster. The learned models were used as generic shape priors for circular cells and for segmentation of lungs in X-ray chest radiographs, where it outperformed the GrabCut baseline.

Finally, we proposed a novel method for joint segmentation, detection and tracking of general objects. The method is based on a probabilistic model that is defined implicitly in terms of a Markov chain Monte Carlo (MCMC) algorithm. It contains a temporal feedback, which allows to dynamically alter detector parameters using hints given by neighboring frames and, in this way, correct detection errors that would otherwise propagate into the data association module. This reduces need for a complex and difficult to design detection module and helps the method to achieve competitive results using a simple general purpose detector. The parameters of the model are learned using an objective based on empirical risk minimization. The performance of the method was evaulated on selected datasets from the ISBI Cell Tracking Challenge, where it consistently belonged among the best three methods and it was also used for large-scale experiments with confluent cultures of endothelial cells that investigated the effect of various biochemical treatments to cell elongation and migration ability.

The proposed methods can be extended in various ways. For example, the initial method for tracking dimensionless particles would benefit from automatic parameter learning. Modeling capacity of the probabilistic shape prior could be increased by introducing lateral edges between latent variables or by adding more hidden layers. The final tracking method can be improved in several ways too. It would be straightforward to extend the transition probabilities by terms, that depend on mutual interactions of multiple objects. We could also enlarge the set of detector parameters, that are subject of the temporal feedback and test the method with more types of detectors. The method could also benefit from more elaborate inference algorithm, which would be able to find better local optimum than the greedy algorithm.

Bibliography

[1] Adams, R., Bischof, L.: Seeded region growing. IEEE Trans. Pattern Anal. Mach. Intell. 16(6), 641–647 (Jun 1994), http://dx.doi.org/10.1109/34.295913

[2] Agn, M., Puonti, O., Rosenschöld, P.M.a., Law, I., Van Leemput, K.: Brain tumor segmentation using a generative model with an rbm prior on tumor shape. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Handels, H. (eds.) Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries. pp. 168–180. Springer International Publishing, Cham (2016)

[3] Akram, S.U., Kannala, J., Eklund, L., Heikkilä, J.: Joint cell segmentation and tracking using cell proposals. In: 13th IEEE International Symposium on Biomedical Imaging, ISBI 2016, Prague, Czech Republic, April 13-16, 2016. pp. 920–924 (2016), https://doi.org/10.1109/ISBI.2016.7493415

[4] Al-Faris, A., Ngah, U., Mat Isa, N.A., Shuaib, I.L.: Computer-aided segmentation system for breast mri tumour using modified automatic seeded region growing (bmri-masrg) 27 (10 2013)

[5] Al-Kofahi, Y., Lassoued, W., Lee, W., Roysam, B.: Improved automatic detection and segmentation of cell nuclei in histopathology images. IEEE Transactions on Biomedical Engineering 57(4), 841–852 (2010)

[6] Alex, V., Safwan K. P., M., Chennamsetty, S.S., Krishnamurthi, G.: Generative adversarial networks for brain lesion detection. vol. 10133, pp. 10133 – 10133 – 9 (2017), https://doi.org/10.1117/12.2254487

[7] Araújo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., Polónia, A., Campilho, A.: Classification of breast cancer histology images using convolutional neural networks. PLoS One 12(6), e0177544 (Jun 2017), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5453426/, pONE-D-16-48431[PII]

[8] Arce, S.H., Wu, P.H., Tseng, Y.: Fast and accurate automated cell boundary determination for fluorescence microscopy. Scientific Reports

3, 2266 EP – (Jul 2013), http://dx.doi.org/10.1038/srep02266, article

[9] Asari, V.K., Sankaran, P.: Adaptive thresholding based cell segmentation for cell-destruction activity verification. In: 35th Applied Imagery Pattern Recognition Workshop (AIPR 2006)(AIPR). vol. 00, p. 14 (10 2006), doi.ieeecomputersociety.org/10.1109/AIPR.2006.9

[10] Azar, F.S., Metaxas, D., Schnall, M.D.: A deformable finite element model of the breast for predicting mechanical deformations under external perturbations 8, 965–75 (11 2001)

[11] Baloch, S., Cheng, E., Fang, T.: Shape Based Conditional Random Fields for Segmenting Intracranial Aneurysms, pp. 55–67. Springer Netherlands, Dordrecht (2013), https://doi.org/10.1007/978-94-007-4255-0_4

[12] Bauer, S., Nolte, L.P., Reyes, M.: Fully automatic segmentation of brain tumor images using support vector machine classification in combination with hierarchical conditional random field regularization. In: Fichtinger, G., Martel, A., Peters, T. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011. pp. 354–361. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[13] Bauer, S., Wiest, R., Nolte, L.P., Reyes, M.: A survey of mri-based medical image analysis for brain tumor studies. Physics in medicine and biology 58 13, R97–129 (2013)

[14] Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. The Annals of Mathematical Statistics 41(1), 164–171 (1970), http://dx.doi.org/10.2307/2239727

[15] Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M.: Automatic differentiation in machine learning: a survey. Journal of Machine Learning Research 18(153), 1–43 (2018), http://jmlr.org/papers/v18/17-468.html

[16] Bengio, Y.: Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1), 1–127 (2009)

[17] Bensch, R., Ronneberger, O.: Cell segmentation and tracking in phase contrast images using graph cut with asymmetric boundary costs. In: 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI). pp. 1220–1223 (2015)

[18] Berclaz, J., Fleuret, F., Turetken, E., Fua, P.: Multiple object tracking using k-shortest paths optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(9), 1806–1819 (2011)

[19] Besag, J.: Spatial Interaction and the Statistical Analysis of Lattice Systems. Journal of the Royal Statistical Society. Series B (Methodological) 36(2), 192–236 (1974), http://dx.doi.org/10.2307/2984812

[20] Besag, J.: On the statistical-analysis of dirty pictures B-48, 259–302 (01 1986)

[21] Beucher, S., Lantuéjoul, C.: Use of watersheds in contour detection 132 (01 1979)

[22] Beucher, S., Meyer, F.: The morphological approach to segmentation: The watershed transformation Vol. 34, 433–481 (01 1993)

[23] Bogovic, J.A., Prince, J.L., Bazin, P.L.: A multiple object geometric deformable model for image segmentation. Comput Vis Image Underst 117(2), 145–157 (Feb 2013), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3539759/, 23316110[pmid]

[24] Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(9), 1124–1137 (2004)

[25] Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(11), 1222–1239 (2001)

[26] Bradley, D., Roth, G.: Adaptive thresholding using the integral image 12, 13–21 (01 2007)

[27] Bulatov, A.A., Grohe, M.: The complexity of partition functions. Theor. Comput. Sci. 348(2-3), 148–186 (2005)

[28] Busygin, S.: A new trust region technique for the maximum weight clique problem. Discrete Applied Mathematics 154(15), 2080 – 2096 (2006), http://www.sciencedirect.com/science/article/pii/S0166218X06000497, international Symposium on Combinatorial Optimization CO'02

[29] Cao, J., Ehling, M., März, S., Seebach, J., Tarbashevich, K., Sixta, T., Pitulescu, M.E., Werner, A.C., Flach, B., Montanez, E., Raz, E., Adams, R.H., Schnittler, H.: Polarized actin and ve-cadherin dynamics regulate junctional remodelling and cell migration during sprouting angiogenesis. Nature Communications 8(1), 2210–2230 (dec 2017), https://doi.org/10.1038/s41467-017-02373-8

[30] Casamitjana, A., Puch, S., Aduriz, A., Vilaplana, V.: 3d convolutional neural networks for brain tumor segmentation: A comparison of multi-resolution architectures. In: Crimi, A., Menze, B., Maier, O., Reyes, M., Winzeck, S., Handels, H. (eds.) Brainlesion: Glioma, Multiple

Sclerosis, Stroke and Traumatic Brain Injuries. pp. 150–161. Springer International Publishing, Cham (2016)

[31] Caselles, V., Catté, F., Coll, T., Dibos, F.: A geometric model for active contours in image processing. Numerische Mathematik 66(1), 1–31 (Dec 1993), https://doi.org/10.1007/BF01385685

[32] Chakraborty, A., Roy-Chowdhury, A.: A conditional random field model for tracking in densely packed cell structures. In: 2014 IEEE International Conference on Image Processing (ICIP). pp. 451–455 (2014)

[33] Chang, H., Han, J., Borowsky, A., Loss, L.A., Gray, J.W., Spellman, P.T., Parvin, B.: Invariant delineation of nuclear architecture in glioblastoma multiforme for clinical and molecular association. IEEE Transactions on Medical Imaging 32, 670–682 (2013)

[34] Chang, H., Parvin, B., Berkeley, L.: Nuclear segmentation in h&e sections via multi-reference graph cut (mrgc) (2011)

[35] Chen, F., Yu, H., Hu, R., Zeng, X.: Deep learning shape priors for object segmentation. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1870–1877 (2013)

[36] Cheng, Y., Wang, D., Zhou, P., Zhang, T.: A survey of model compression and acceleration for deep neural networks. CoRR abs/1710.09282 (2017)

[37] Choi, Y., El-Khamy, M., Lee, J.: Universal deep neural network compression. CoRR abs/1802.02271 (2018)

[38] Cohen, L.D.: On active contour models and balloons. CVGIP: Image Understanding 53(2), 211 – 218 (1991), http://www.sciencedirect.com/science/article/pii/104996609190028N

[39] Cole, R.: Live-cell imaging. Cell Adhesion & Migration 8(5), 452–459 (2014), https://doi.org/10.4161/cam.28348, pMID: 25482523

[40] Cooper, G.F.: The computational complexity of probabilistic inference using bayesian belief networks (research note). Artif. Intell. 42(2-3), 393–405 (Mar 1990), http://dx.doi.org/10.1016/0004-3702(90)90060-D

[41] Cousty, J., Bertrand, G., Najman, L., Couprie, M.: Watershed cuts: Minimum spanning forests and the drop of water principle 31, 1362–74 (09 2009)

[42] Dahl, A., Larsen, R.: Learning dictionaries of discriminative image patches. In: Proceedings of the British Machine Vision Conference. pp. 77.1–77.11. BMVA Press (2011), http://dx.doi.org/10.5244/C.25.77

[43] Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiway cuts (extended abstract). In: Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing. pp. 241–251. STOC '92, ACM, New York, NY, USA (1992), `http://doi.acm.org/10.1145/129712.129736`

[44] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, pp. 886–893 vol. 1 (2005)

[45] Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B 39, 1–38 (1977), `http://web.mit.edu/6.435/www/Dempster77.pdf`

[46] Denil, M., Shakibi, B., Dinh, L., Ranzato, M., de Freitas, N.: Predicting parameters in deep learning. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2. pp. 2148–2156. NIPS'13, Curran Associates Inc., USA (2013), `http://dl.acm.org/citation.cfm?id=2999792.2999852`

[47] Dima, A., Elliott, J., Filliben, J., Halter, M., Peskin, A., Bernal, J., Kociolek, M., Brady, M.C., Tang, H.C., Plant, A.: Comparison of segmentation algorithms for fluorescence microscopy images of cells 79, 545–59 (07 2011)

[48] Ding, Y., Huang, J.Z., Ji, J.X., Park, C.: Segmentation, inference and classification of partially overlapping nanoparticles. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1 (03 2013), `doi.ieeecomputersociety.org/10.1109/TPAMI.2012.163`

[49] Dora, L., Agrawal, S., Panda, R., Abraham, A.: State-of-the-art methods for brain tissue segmentation: A review. IEEE Reviews in Biomedical Engineering 10, 235–249 (2017)

[50] Dufour, A., Shinin, V., Tajbakhsh, S., Guillén-Aghion, N., Olivo-Marin, J.C., Zimmer, C.: Segmenting and tracking fluorescent cells in dynamic 3-d microscopy with coupled active surfaces 14, 1396–410 (10 2005)

[51] Dzyubachyk, O., Cappellen, W.A.v., Essers, J., Niessen, W.J., Meijering, E.H.W.: Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. IEEE Transactions on Medical Imaging 29, 852–867 (2010)

[52] Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. J. ACM 19(2), 248–264 (Apr 1972), `http://doi.acm.org/10.1145/321694.321699`

[53] El-Baz, A., Gimel'farb, G.: Image segmentation with a parametric deformable model using shape and appearance priors. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008)

[54] Eldan, R., Shamir, O.: The power of depth for feedforward neural networks. In: Conference on Learning Theory (12 2015)

[55] Eslami, S.M.A., Heess, N., Williams, C.K.I., Winn, J.: The shape boltzmann machine: A strong model of object shape. International Journal of Computer Vision 107(2), 155–176 (Apr 2014), https://doi.org/10.1007/s11263-013-0669-1

[56] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature 542, 115 EP – (Jan 2017), http://dx.doi.org/10.1038/nature21056

[57] Farag, A.: Deformable Models: Theory and Biomaterial Applications. Topics in Biomedical Engineering. International Book Series, Springer New York (2007), https://books.google.cz/books?id=Yd92tQEACAAJ

[58] Fiaschi, L., Diego, F., Gregor, K., Schiegg, M., Koethe, U., Zlatic, M., Hamprecht, F.A.: Tracking indistinguishable translucent objects over time using weakly supervised structured learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2014)

[59] Fix, A., Chen, J., Boros, E., Zabih, R.: Approximate mrf inference using bounded treewidth subgraphs. In: Proceedings of the 12th European conference on Computer Vision - Volume Part I. pp. 385–398. ECCV'12, Springer-Verlag, Berlin, Heidelberg (2012), http://dx.doi.org/10.1007/978-3-642-33718-5_28

[60] Flach, B.: A class of random fields on complete graphs with tractable partition function. IEEE Transactions on Pattern Analysis and Machine Intelligence 99(PrePrints), 1 (2013)

[61] Fortmann, T., Bar-Shalom, Y., Scheffe, M.: Sonar tracking of multiple targets using joint probabilistic data association. IEEE Journal of Oceanic Engineering 8(3), 173–184 (1983)

[62] Franz, C.M., Jones, G.E., Ridley, A.J.: Cell migration in development and disease. Developmental Cell 2(2), 153–158 (Feb 2002), http://dx.doi.org/10.1016/S1534-5807(02)00120-X

[63] Fuchs, T.J., Buhmann, J.M.: Computational pathology: Challenges and promises for tissue analysis. Computerized Medical Imaging and Graphics 35(7), 515 – 530 (2011), http://www.sciencedirect.com/

science/article/pii/S0895611111000383, whole Slide Image Process

[64] Geiger, D., Girosi, F.: Parallel and deterministic algorithms from mrfs: Surface reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. 13(5), 401–412 (May 1991), http://dx.doi.org/10.1109/34.134040

[65] Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. 6(6), 721–741 (Nov 1984), https://doi.org/10.1109/TPAMI.1984.4767596

[66] Geremia, E., Menze, B.H., Prastawa, M., Weber, M.A., Criminisi, A., Ayache, N.: Brain tumor cell density estimation from multi-modal mr images based on a synthetic tumor growth model. In: Menze, B.H., Langs, G., Lu, L., Montillo, A., Tu, Z., Criminisi, A. (eds.) Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging. pp. 273–282. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)

[67] Ghanei, A., Soltanian-Zadeh, H., Ratkewicz, A., Yin, F.F.: A three-dimensional deformable model for segmentation of human prostate from ultrasound image 28, 2147–53 (11 2001)

[68] Ghaye, J., Kamat, M.A., Corbino-Giunta, L., Silacci, P., Vergères, G., Micheli, G., Carrara, S.: Image thresholding techniques for localization of sub-resolution fluorescent biomarkers. Cytometry Part A 83(11) (2013)

[69] Godinez, W.J., Rohr, K.: Tracking multiple particles in fluorescence time-lapse microscopy images via probabilistic data association. IEEE Transactions on Medical Imaging 34, 415–432 (2015)

[70] Goldberg, A.V.: An efficient implementation of a scaling minimum-cost flow algorithm. Journal of Algorithms 22(1), 1 – 29 (1997), http://www.sciencedirect.com/science/article/pii/S019667748570805X

[71] Goulian, M., Simon, S.M.: Tracking single proteins within cells. Biophysical Journal 79(4), 2188 – 2198 (2000), http://www.sciencedirect.com/science/article/pii/S0006349500764678

[72] Hammersley, J.M., Clifford, P.E.: Markov random fields on finite graphs and lattices. *Unpublished manuscript* (1971)

[73] Hara, K., Saitoh, D., Shouno, H.: Analysis of dropout learning regarded as ensemble learning. CoRR abs/1706.06859 (2017)

[74] Hatt, M., Laurent, B., Ouahabi, A., Fayad, H., Tan, S., Li, L., Lu, W., Jaouen, V., Tauber, C., Czakon, J., Drapejkowski, F., Dyrka, W., Camarasu-Pop, S., Cervenansky, F., Girard, P., Glatard, T., Kain,

M., Yao, Y., Barillot, C., Kirov, A., Visvikis, D.: The first miccai challenge on pet tumor segmentation. Medical Image Analysis 44, 177 – 195 (2018), http://www.sciencedirect.com/science/article/pii/S1361841517301895

[75] Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.M., Larochelle, H.: Brain tumor segmentation with deep neural networks. Medical Image Analysis 35, 18 – 31 (2017), http://www.sciencedirect.com/science/article/pii/S1361841516300330

[76] He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. CoRR abs/1703.06870 (2017)

[77] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR abs/1512.03385 (2015)

[78] Heinemann, U., Globerson, A.: What cannot be learned with Bethe approximations. In: UAI. pp. 319–326 (2011)

[79] Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14, 1771–1800 (2002)

[80] Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. 18(7), 1527–1554 (Jul 2006), http://dx.doi.org/10.1162/neco.2006.18.7.1527

[81] Hodneland, E., Bukoreshtliev, N.V., Eichler, T.W., Tai, X., Gurke, S., Lundervold, A., Gerdes, H.: A unified framework for automated 3-d segmentation of surface-stained living cells and a comprehensive segmentation evaluation. IEEE Trans. Med. Imaging 28(5), 720–738 (2009), https://doi.org/10.1109/TMI.2008.2011522

[82] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural Networks 4(2), 251 – 257 (1991), http://www.sciencedirect.com/science/article/pii/089360809190009T

[83] Hou, L., Samaras, D., Kurc, T.M., Gao, Y., Davis, J.E., Saltz, J.H.: Patch-based convolutional neural network for whole slide tissue image classification. Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit 2016, 2424–2433 (2016), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC5085270/, 27795661[pmid]

[84] Huh, S., Chen, M.: Detection of mitosis within a stem cell population of high cell confluence in phase-contrast microscopy images. In: CVPR 2011. pp. 1033–1040 (2011)

[85] Ihler, A.T., Fisher III, J.W., Willsky, A.S., Chickering, M.: Loopy belief propagation: Convergence and effects of message errors. Journal of Machine Learning Research 6, 905–936 (2005)

[86] Irshad, H., Veillard, A., Roux, L., Racoceanu, D.: Methods for nuclei detection, segmentation and classification in digital histopathology: A review current status and future potential 7, 97–114 (05 2014)

[87] Jamil, N., Soh, H.C., Tengku Sembok, T.M., Bakar, Z.A.: A modified edge-based region growing segmentation of geometric objects. In: Badioze Zaman, H., Robinson, P., Petrou, M., Olivier, P., Shih, T.K., Velastin, S., Nyström, I. (eds.) Visual Informatics: Sustaining Research and Innovations. pp. 99–112. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[88] Jaward, M., Mihaylova, L., Canagarajah, N., Bull, D.: Multiple object tracking using particle filters. In: 2006 IEEE Aerospace Conference. p. 8 pp. (2006)

[89] Jensen, E.C.: Overview of live-cell imaging: Requirements and methods used. The Anatomical Record 296(1), 1–8 (2013), https://onlinelibrary.wiley.com/doi/abs/10.1002/ar.22554

[90] John, J., Nair, M.S., Kumar, P.A., Wilscy, M.: A novel approach for detection and delineation of cell nuclei using feature similarity index measure. Biocybernetics and Biomedical Engineering 36(1), 76 – 88 (2016), http://www.sciencedirect.com/science/article/pii/S0208521615000856

[91] K. Justice, R., Stokely, E., Strobel, J., E. Ideker, R., M. Smith, W.: Medical image segmentation using 3d seeded region growing 3034 (04 1997)

[92] Kae, A., Sohn, K., Lee, H., Learned-Miller, E.G.: Augmenting crfs with boltzmann machine shape priors for image labeling. 2013 IEEE Conference on Computer Vision and Pattern Recognition pp. 2019–2026 (2013)

[93] Kainz, P., Urschler, M., Schulter, S., Wohlhart, P., Lepetit, V.: You should use regression to detect cells. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 276–283. Springer International Publishing, Cham (2015)

[94] Kalaidzidis, Y.: Multiple objects tracking in fluorescence microscopy. J Math Biol 58(1-2), 57–80 (Jan 2009), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2798991/, 180[PII]

[95] Kapur, J., Sahoo, P., Wong, A.: A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, and Image Processing 29(3), 273 – 285 (1985), http://www.sciencedirect.com/science/article/pii/0734189X85901252

[96] Karimaghaloo, Z., Arnold, D.L., Arbel, T.: Adaptive multi-level conditional random fields for detection and segmentation of small enhanced pathology in medical images. Medical Image Analysis 27, 17 – 30 (2016), http://www.sciencedirect.com/science/article/pii/S1361841515000912, discrete Graphical Models in Biomedical Image Analysis

[97] Karp, G.: Techniques in Cell and Molecular Biology, pp. 715–762. Wiley (2009)

[98] Karsch, K., He, Q., Duan, Y.: A fast, semi-automatic brain structure segmentation algorithm for magnetic resonance imaging. 2009 IEEE International Conference on Bioinformatics and Biomedicine pp. 297–302 (2009)

[99] Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1(4), 321–331 (Jan 1988), https://doi.org/10.1007/BF00133570

[100] Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). pp. 4696–4704. ICCV '15, IEEE Computer Society, Washington, DC, USA (2015), http://dx.doi.org/10.1109/ICCV.2015.533

[101] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR abs/1412.6980 (2014)

[102] Kohli, P., Kumar, M.P., Torr, P.H.S.: P3 & beyond: Solving energies with higher order cliques. In: CVPR (2007)

[103] Kolmogorov, V.: Convergent tree-reweighted message passing for energy minimization. IEEE Trans. Pattern Anal. Mach. Intell. 28(10), 1568–1583 (Oct 2006), http://dx.doi.org/10.1109/TPAMI.2006.200

[104] Kong, H., N. Gurcan, M., Belkacem-Boussaid, K.: Partitioning histopathological images: An integrated framework for supervised color-texture segmentation and cell splitting 30, 1661–77 (04 2011)

[105] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[106] Kuhn, H.W.: The Hungarian Method for the Assignment Problem, pp. 29–47. Springer Berlin Heidelberg, Berlin, Heidelberg (2010), https://doi.org/10.1007/978-3-540-68279-0_2

[107] Kurokawa, H., Noda, H., Sugiyama, M., Sakaue-Sawano, A., Fukami, K., Miyawaki, A.: Software for precise tracking of cell proliferation. Biochemical and Biophysical Research Communications 417(3), 1080 – 1085 (2012), http://www.sciencedirect.com/science/article/pii/S0006291X11023096

[108] Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning. pp. 282–289. ICML '01, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2001), http://dl.acm.org/citation.cfm?id=645530.655813

[109] LeCun, Y., Bengio, Y.: The handbook of brain theory and neural networks. chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. MIT Press, Cambridge, MA, USA (1998), http://dl.acm.org/citation.cfm?id=303568.303704

[110] Li, K., Miller, E.D., Chen, M., Kanade, T., Weiss, L.E., Campbell, P.G.: Cell population tracking and lineage construction with spatiotemporal context. Medical Image Analysis 12(5), 546 – 566 (2008), http://www.sciencedirect.com/science/article/pii/S1361841508000650, special issue on the 10th international conference on medical imaging and computer assisted intervention - MICCAI 2007

[111] Li, M., Castillo, E., Zheng, X.L., Luo, H.Y., Castillo, R., Wu, Y., Guerrero, T.: Modeling lung deformation: A combined deformable image registration method with spatially varying young's modulus estimates 40, 081902 (08 2013)

[112] Liang, S., Srikant, R.: Why deep neural networks? CoRR abs/1610.04161 (2016)

[113] Lindsay, B.G.: Composite Likelihood Methods. Contemporary Mathematics 80, 221–239 (1988)

[114] Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1150–1157 vol.2 (1999)

[115] Luo, S., Li, X., Li, J.: Review on the methods of automatic liver segmentation from abdominal images 02, 1–7 (01 2014)

[116] Luo, W., Zhao, X., Kim, T.: Multiple object tracking: A review. CoRR abs/1409.7618 (2014)

[117] Ma, Z., Tavares, J.M.R., Jorge, R.N., Mascarenhas, T.: A review of algorithms for medical image segmentation and their applications to the female pelvic cavity. Computer Methods in Biomechanics and

Biomedical Engineering 13(2), 235–246 (2010), https://doi.org/10.1080/10255840903131878

[118] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. pp. 281–297. University of California Press, Berkeley, Calif. (1967), https://projecteuclid.org/euclid.bsmsp/1200512992

[119] Magnusson, K., Jalden, J., M. Gilbert, P., Blau, H.: Global linking of cell tracks using the viterbi algorithm 34 (11 2014)

[120] Makni, N., Puech, P., Lopes, R., Dewalle, A.S., Colot, O., Betrouni, N.: Combining a deformable model and a probabilistic framework for an automatic 3d segmentation of prostate on mri. International Journal of Computer Assisted Radiology and Surgery 4(2), 181 (Dec 2008), https://doi.org/10.1007/s11548-008-0281-y

[121] Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: a level set approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(2), 158–175 (1995)

[122] Marlin, B., Swersky, K., Chen, B., de Freitas, N.: Inductive principles for restricted Boltzmann machine learning. Journal of Machine Learning Research 9, 509–516 (2010)

[123] Martin, S., Troccaz, J., Daanen, V.: Automated segmentation of the prostate in 3d mr images using a probabilistic atlas and a spatially constrained deformable model. Medical Physics 37(4), 1579–1590 (2010), https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3315367

[124] Massich, J., Meriaudeau, F., Sentís, M., Ganau, S., Pérez, E., Martí, R., Oliver, A., Martí, J.: Automatic seed placement for breast lesion segmentation on us images. In: Maidment, A.D.A., Bakic, P.R., Gavenonis, S. (eds.) Breast Imaging. pp. 308–315. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)

[125] Maška, M., Ulman, V., Svoboda, D., Matula, P., Matula, P., Ederra, C., Urbiola, A., España, T., Venkatesan, S., Balak, D.M., Karas, P., Bolcková, T., Štreitová, M., Carthel, C., Coraluppi, S., Harder, N., Rohr, K., Magnusson, K.E.G., Jaldén, J., Blau, H.M., Dzyubachyk, O., Křížek, P., Hagen, G.M., Pastor-Escuredo, D., Jimenez-Carretero, D., Ledesma-Carbayo, M.J., Muñoz-Barrutia, A., Meijering, E., Kozubek, M., Ortiz-de Solorzano, C.: A benchmark for comparison of cell tracking algorithms. Bioinformatics 30(11), 1609–1617 (2014), +http://dx.doi.org/10.1093/bioinformatics/btu080

[126] McInerney, T., Terzopoulos, D.: Deformable models in medical image analysis: a survey. Medical Image Analysis 1(2), 91 –

108 (1996), http://www.sciencedirect.com/science/article/pii/S1361841596800077

[127] Meier, U., López, O., Monserrat, C., Juan, M.C., Alcañiz, M.: Real-time deformable models for surgery simulation: A survey. Comput. Methods Prog. Biomed. 77(3), 183–197 (Mar 2005), http://dx.doi.org/10.1016/j.cmpb.2004.11.002

[128] Meijering, E.: Cell segmentation: 50 years down the road [life sciences]. IEEE Signal Process. Mag. 29(5), 140–145 (2012), http://dblp.uni-trier.de/db/journals/spm/spm29.html#Meijering12

[129] Meijster, A., Roerdink, J.B.T.M., Hesselink, W.H.: Mathematical Morphology and its Applications to Image and Signal Processing, chap. A General Algorithm for Computing Distance Transforms in Linear Time, pp. 331–340. Springer US, Boston, MA (2000)

[130] Melouah, A.: Comparison of automatic seed generation methods for breast tumor detection using region growing technique. In: Amine, A., Bellatreche, L., Elberrichi, Z., Neuhold, E.J., Wrembel, R. (eds.) Computer Science and Its Applications. pp. 119–128. Springer International Publishing, Cham (2015)

[131] Memariani, A., Nikou, C., Endres, B., Bassères, E., Garey, K., Kakadiaris, I.: Detcic: Detection of elongated touching cells with inhomogeneous illumination using a stack of conditional random fields pp. 574–580 (01 2018)

[132] Mhaskar, H., Liao, Q., Poggio, T.: When and why are deep networks better than shallow ones? (2017), https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14849

[133] Niblack, W.: An Introduction to Digital Image Processing. Strandberg Publishing Company, Birkeroed, Denmark, Denmark (1985)

[134] Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition 29(1), 51 – 59 (1996), http://www.sciencedirect.com/science/article/pii/0031320395000674

[135] Olivo-Marin, J.C., Meijering, E.: Objective comparison of particle tracking methods. Nature Methods 11(3), 281–289 (2014)

[136] Ososkov, G., Goncharov, P.: Shallow and deep learning for image classification. Optical Memory and Neural Networks 26(4), 221–248 (Oct 2017), https://doi.org/10.3103/S1060992X1704004X

[137] Otsu, N.: A Threshold Selection Method from Gray-level Histograms. IEEE Transactions on Systems, Man and Cybernetics 9(1), 62–66 (1979), http://dx.doi.org/10.1109/TSMC.1979.4310076

[138] Palumbo, P.W., Swaminathan, P., Srihari, S.N.: Document image binarization: Evaluation of algorithms. vol. 0697, pp. 0697 – 0697 – 8 (1986), https://doi.org/10.1117/12.976229

[139] Pan, J., Kanade, T., Chen, M.: Heterogeneous conditional random field: Realizing joint detection and segmentation of cell regions in microscopic images. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2940–2947 (2010)

[140] Papageorgiou, D.J., Salpukas, M.R.: The maximum weight independent set problem for data association in multiple hypothesis tracking. In: Hirsch, M.J., Commander, C.W., Pardalos, P.M., Murphey, R. (eds.) Optimization and Cooperative Control Strategies. pp. 235–255. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)

[141] Pearl, J.: Reverend bayes on inference engines: A distributed hierarchical approach. In: Proceedings of the Second AAAI Conference on Artificial Intelligence. pp. 133–136. AAAI'82, AAAI Press (1982), http://dl.acm.org/citation.cfm?id=2876686.2876719

[142] Pecot, T., Chessel, A., Bardin, S., Salamero, J., Bouthemy, P., Kervrann, C.: Conditional random fields for object and background estimation in fluorescence video-microscopy. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. pp. 734–737 (2009)

[143] Pellegrini, S., Ess, A., Van Gool, L.: Improving data association by joint modeling of pedestrian trajectories and groupings. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) Computer Vision – ECCV 2010. pp. 452–465. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)

[144] Perez, L., Wang, J.: The effectiveness of data augmentation in image classification using deep learning. CoRR abs/1712.04621 (2017)

[145] Pham, D.L., Xu, C., Prince, J.L.: Image segmentation using deformable models. In: Sonka, M., Fitzpatrick, J., R. Masters, B. (eds.) Handbook of Medical Imaging, Volume 2: Medical Image Processing and Analysis, chap. 3 (01 2002)

[146] Piccinini, F., Kiss, A., Horvath, P.: Celltracker (not only) for dummies. Bioinformatics 32(6), 955–957 (2016), +http://dx.doi.org/10.1093/bioinformatics/btv686

[147] Pirsiavash, H., Ramanan, D., Fowlkes, C.C.: Globally-optimal greedy algorithms for tracking a variable number of objects. In: CVPR 2011. pp. 1201–1208 (2011)

[148] Pohle, R., Tönnies, K.: Segmentation of medical images using adaptive region growing 4322 (01 2002)

[149] Qin, Z.: Improving multi-target tracking via social grouping. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1972–1978. CVPR '12, IEEE Computer Society, Washington, DC, USA (2012), http://dl.acm.org/citation.cfm?id=2354409.2354969

[150] Rao, V., Teh, Y.W.: Fast MCMC sampling for Markov jump processes and extensions. Journal of Machine Learning Research 14, 3207–3232 (2013), arXiv:1208.4818

[151] Ray, N., Acton, S.T., Ley, K.: Tracking leukocytes in vivo with shape and size constrained active contours. IEEE Transactions on Medical Imaging 21(10), 1222–1235 (2002)

[152] Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. CoRR abs/1506.02640 (2015)

[153] Reid, D.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 24(6), 843–854 (1979)

[154] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1. pp. 91–99. NIPS'15, MIT Press, Cambridge, MA, USA (2015), http://dl.acm.org/citation.cfm?id=2969239.2969250

[155] Rezatofighi, S.H., Gould, S., Hartley, R., Mele, K., Hughes, W.E.: Application of the imm-jpda filter to multiple target tracking in total internal reflection fluorescence microscopy images. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012. pp. 357–364. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)

[156] Rezatofighi, S.H., Milan, A., Zhang, Z., Shi, Q., Dick, A., Reid, I.: Joint probabilistic data association revisited. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). pp. 3047–3055. ICCV '15, IEEE Computer Society, Washington, DC, USA (2015), http://dx.doi.org/10.1109/ICCV.2015.349

[157] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)

[158] Rosin, P.L.: Measuring shape: ellipticity, rectangularity, and triangularity. Machine Vision and Applications 14(3), 172–184 (2003), http://dx.doi.org/10.1007/s00138-002-0118-6

[159] Roth, H.R., Lu, L., Farag, A., Sohn, A., Summers, R.M.: Spatial aggregation of holistically-nested networks for automated pancreas segmentation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. pp. 451–459. Springer International Publishing, Cham (2016)

[160] Roth, S., Black, M.J.: Fields of experts. International Journal of Computer Vision 82(2), 205 (Jan 2009), https://doi.org/10.1007/s11263-008-0197-6

[161] Rother, C., Kolmogorov, V., Blake, A.: GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. 23(3), 309–314 (2004)

[162] Roullier, V., Lezoray, O., Ta, V., Elmoataz, A.: Multi-resolution graph-based analysis of histopathological whole slide images: Application to mitotic cell extraction and visualization 35, 603–15 (05 2011)

[163] Rue, H., Syversveen, A.R.: Bayesian object recognition with baddeley's delta loss. Adv. in Appl. Probab. 30(1), 64–84 (03 1998)

[164] Ruggeri, M., Tsechpenakis, G., Jiao, S., Elena Jockovich, M., Cebulla, C., Hernandez, E., Murray, T., A Puliafito, C.: Retinal tumor imaging and volume quantification in mouse model using spectral-domain optical coherence tomography 17, 4074–83 (04 2009)

[165] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representation by error propagation. In: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. Vol. 1 (01 1986)

[166] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015)

[167] Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: van Dyk, D., Welling, M. (eds.) Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 5, pp. 448–455. PMLR, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA (16–18 Apr 2009), http://proceedings.mlr.press/v5/salakhutdinov09a.html

[168] Schiegg, M., Hanslovsky, P., Haubold, C., Köthe, U., Hufnagel, L., Hamprecht, F.A.: Graphical model for joint segmentation and tracking of multiple dividing cell. Bioinformatics 31(6), 948–956 (2015), http://bioinformatics.oxfordjournals.org/content/early/2014/11/17/bioinformatics.btu764.full.pdf?keytype=ref&ijkey=mTXWsiFrci7R8tc, 1

[169] Schlesinger, M.I.: The interaction of learning and self-organization in pattern recognition. Cybernetics and Systems Analysis 4(2), 66–71 (1968)

[170] Schmid, J., Magnenat-Thalmann, N.: Mri bone segmentation using deformable models and shape priors. In: Metaxas, D., Axel, L., Fichtinger, G., Székely, G. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2008. pp. 119–126. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)

[171] Schmidt-Richberg, A., Brosch, T., Schadewaldt, N., Klinder, T., Cavallaro, A., Salim, I., Roundhill, D., Papageorghiou, A., Lorenz, C.: Abdomen segmentation in 3d fetal ultrasound using cnn-powered deformable models. In: Cardoso, M.J., Arbel, T., Melbourne, A., Bogunovic, H., Moeskops, P., Chen, X., Schwartz, E., Garvin, M., Robinson, E., Trucco, E., Ebner, M., Xu, Y., Makropoulos, A., Desjardin, A., Vercauteren, T. (eds.) Fetal, Infant and Ophthalmic Medical Image Analysis. pp. 52–61. Springer International Publishing, Cham (2017)

[172] Sethi, I.K., Jain, R.: Finding trajectories of feature points in a monocular image sequence. IEEE Trans. Pattern Anal. Mach. Intell. 9(1), 56–73 (Jan 1987), https://doi.org/10.1109/TPAMI.1987.4767872

[173] Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences of the United States of America 93(4), 1591–1595 (1996), http://www.jstor.org/stable/38628

[174] Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. J. Electronic Imaging 13(1), 146–168 (2004), http://dblp.uni-trier.de/db/journals/jei/jei13.html#SezginS04

[175] Shan, J., Cheng, H.D., Wang, Y.: A novel automatic seed point selection algorithm for breast ultrasound images. 2008 19th International Conference on Pattern Recognition pp. 1–4 (2008)

[176] Shimony, S.E.: Finding maps for belief networks is np-hard. Artificial Intelligence 68(2), 399 – 410 (1994), http://www.sciencedirect.com/science/article/pii/0004370294900728

[177] Shiraishi, J., Katsuragawa, S., Ikezoe, J., Matsumoto, T., Kobayashi, T., Komatsu, K.i., Matsui, M., Fujita, H., Kodera, Y., Doi, K.: Development of a digital image database for chest radiographs with and without a lung nodule. American Journal of Roentgenology 174(1), 71–74 (2000)

[178] Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: Proceedings of the Seventh International Conference on Document Analysis

and Recognition - Volume 2. pp. 958–. ICDAR '03, IEEE Computer Society, Washington, DC, USA (2003), http://dl.acm.org/citation.cfm?id=938980.939477

[179] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556 (2014)

[180] Sohl-Dickstein, J., Battaglino, P., DeWeese, M.R.: Minimum probability flow learning. In: Proceedings of the 28th International Conference on International Conference on Machine Learning. pp. 905–912. ICML'11, Omnipress, USA (2011), http://dl.acm.org/citation.cfm?id=3104482.3104596

[181] Sokal, A.D.: Monte carlo methods in statistical mechanics: Foundations and new algorithms. Lectures notes (1989)

[182] Srinivasan, P., Ravindran, G.: A complete automatic region growing method for segmentation of masses on ultrasound images. In: Intl Conf Biomed Pharm Eng. pp. 88 – 92 (01 2007)

[183] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958 (2014), http://jmlr.org/papers/v15/srivastava14a.html

[184] Sung, M.H., McNally, J.G.: Live cell imaging and systems biology. Wiley Interdiscip Rev Syst Biol Med 3(2), 167–182 (Aug 2011), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2992103/, 20730797[pmid]

[185] Suurballe, J.W.: Disjoint paths in a network. Networks 4(2), 125–145 (1974), https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230040204

[186] Swedlow, J.R.: Innovation in biological microscopy: Current status and future directions. Bioessays 34(5), 333–340 (May 2012), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3427900/, 22408015[pmid]

[187] Terzopoulos, D., Witkin, A., Kass, M.: Constraints on deformable models:recovering 3d shape and nonrigid motion. Artificial Intelligence 36(1), 91 – 123 (1988), http://www.sciencedirect.com/science/article/pii/000437028890080X

[188] Tieleman, T.: Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. In: Proceedings of the 25th international conference on Machine learning. pp. 1064–1071. ACM New York, NY, USA (2008)

[189] Tieleman, T., Hinton, G.: Using Fast Weights to Improve Persistent Contrastive Divergence. In: Proceedings of the 26th international conference on Machine learning. pp. 1033–1040. ACM New York, NY, USA (2009)

[190] Tinevez, J.Y., Perry, N., Schindelin, J., Hoopes, G.M., Reynolds, G.D., Laplantine, E., Bednarek, S.Y., Shorte, S.L., Eliceiri, K.W.: Trackmate: An open and extensible platform for single-particle tracking. Methods 115, 80 – 90 (2017), http://www.sciencedirect.com/science/article/pii/S1046202316303346, image Processing for Biologists

[191] Tong, T., Gao, Q.: Extraction of features from patch based graphs for the prediction of disease progression in ad. In: Huang, D.S., Jo, K.H., Hussain, A. (eds.) Intelligent Computing Theories and Methodologies. pp. 500–509. Springer International Publishing, Cham (2015)

[192] Trepat, X., Chen, Z., Jacobson, K.: Cell migration. Compr Physiol 2(4), 2369–2392 (Oct 2012), http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4457291/, 23720251[pmid]

[193] Tsechpenakis, G.: Deformable model-based medical image segmentation. In: Multi Modality State-of-the-Art Medical Image Segmentation and Registration Methodologies, vol. 1, pp. 33–67 (04 2011)

[194] Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) Computer Vision – ECCV 2006. pp. 589–600. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[195] Türetken, E., Wang, X., Becker, C.J., Haubold, C., Fua, P.: Network flow integer programming to track elliptical cells in time-lapse sequences. IEEE Transactions on Medical Imaging 36(4), 942–951 (2017)

[196] Ulman, V., Maska, M., Magnusson, K.E.G., Ronneberger, O., Haubold, C., Harder, N., Matula, P., Matula, P., Svoboda, D., Radojevic, M., Smal, I., Rohr, K., Jaldén, J., Blau, H.M., Dzyubachyk, O., Lelieveldt, B., Xiao, P., Li, Y., Cho, S.Y., Dufour, A.C., Olivo-Marin, J.C., Reyes-Aldasoro, C.C., Solis-Lemus, J.A., Bensch, R., Brox, T., Stegmaier, J., Mikut, R., Wolf, S., Hamprecht, F.A., Esteves, T., Quelhas, P., Demirel, Ö., Malmström, L., Jug, F., Tomancak, P., Meijering, E., Muñoz-Barrutia, A., Kozubek, M., Ortiz-de Solorzano, C.: An objective comparison of cell-tracking algorithms. Nature Methods 14, 1141 EP – (Oct 2017), http://dx.doi.org/10.1038/nmeth.4473

[197] Varin, C., Reid, N., Firth, D.: An Overview of Composite Likelihood Methods. Statistica Sinica 21, 5–42 (2011)

[198] Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inf. Theor. 13(2), 260–269 (Sep 1967), https://doi.org/10.1109/TIT.1967.1054010

[199] Wainwright, M.J., Jaakkola, T.S., Willsky, A.S.: Exact MAP estimates by (hyper)tree agreement. In: NIPS. pp. 809–816. MIT Press (2002)

[200] Wainwright, M.J., Jaakkola, T.S., Willsky, A.S.: Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In: AISTATS. Society for Artificial Intelligence and Statistics (2003)

[201] Welf, E., Driscoll, M., Dean, K., Schäfer, C., Chu, J., Davidson, M., Lin, M., Danuser, G., Fiolka, R.: Quantitative multiscale cell imaging in controlled 3d microenvironments. Developmental Cell 36(4), 462 – 475 (2016), http://www.sciencedirect.com/science/article/pii/S1534580716000897

[202] White, J.M., Rohrer, G.D.: Image thresholding for optical character recognition and other applications requiring character image extraction. IBM J. Res. Dev. 27(4), 400–411 (Jul 1983), http://dx.doi.org/10.1147/rd.274.0400

[203] Wu, J., Poehlman, S., Noseworthy, M., V. Kamath, M.: Texture feature based automated seeded region growing in abdominal mri segmentation 02, 263–267 (06 2008)

[204] Wu, Q., Merchant, F., Castleman, K.: Microscope Image Processing. Academic Press, 1st edn. (2008)

[205] Wu, X., Amrikachi, M., Shah, S.K.: Embedding topic discovery in conditional random fields model for segmenting nuclei using multispectral data. IEEE Transactions on Biomedical Engineering 59(6), 1539–1549 (2012)

[206] Xing, F., Yang, L.: Robust nucleus/cell detection and segmentation in digital pathology and microscopy images: A comprehensive review. IEEE Reviews in Biomedical Engineering 9, 234–263 (2016)

[207] Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing 7(3), 359–369 (1998)

[208] Xu, C., Yezzi, A., Prince, J.L.: On the relationship between parametric and geometric active contours. In: Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No.00CH37154). vol. 1, pp. 483–489 vol.1 (2000)

[209] Yamaguchi, K., Berg, A.C., Ortiz, L.E., Berg, T.L.: Who are you with and where are you going? In: CVPR 2011. pp. 1345–1352 (2011)

[210] Yang, X., Li, H., Zhou, X.: Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy. IEEE Transactions on Circuits and Systems I: Regular Papers 53, 2405–2414 (2006)

[211] Yedidia, J.S., Freeman, W.T., Weiss, Y.: Generalized belief propagation. In: IN NIPS 13. pp. 689–695. MIT Press (2000)

[212] Yin, Z., Kanade, T., Chen, M.: Understanding the phase contrast optics to restore artifact-free microscopy images for segmentation. Medical Image Analysis 16(5), 1047 – 1062 (2012), http://www.sciencedirect.com/science/article/pii/S1361841512000035

[213] Yu, D., Seltzer, M.L., Li, J., Huang, J., Seide, F.: Feature learning in deep neural networks - A study on speech recognition tasks. CoRR abs/1301.3605 (2013)

[214] Yuille, A.L.: Cccp algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. Neural Computation 14, 2002 (2002)

[215] Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8 (2008)

[216] Zhang, S., Huang, J., Uzunbas, M., Shen, T., Delis, F., Huang, X., Volkow, N., Thanos, P., Metaxas, D.N.: 3d segmentation of rodent brain structures using hierarchical shape priors and deformable models. In: Fichtinger, G., Martel, A., Peters, T. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2011. pp. 611–618. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[217] Zhao, H.: Fast sweeping method for eikonal equations 74, 603–627 (04 2005)

[218] Zimmer, C., Labruyere, E., Meas-Yedid, V., Guillen, N., Olivo-Marin, J.C.: Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing. IEEE Transactions on Medical Imaging 21, 1212–1221 (2002)

[219] Zucker, S.W.: Region growing: Childhood and adolescence. Computer Graphics and Image Processing 5(3), 382 – 399 (1976), http://www.sciencedirect.com/science/article/pii/S0146664X76800147

Appendix A

Experiments with Endothelial Cells

E1: Cells transfected with adenoN17Rac virus (adenoN17Rac) vs. cells transfected with adenoempty virus (control)
Group 1: 3 sequences, on average 364 cells in each sequence
Group 2: 3 sequences, 377 cells
Duration: 26 hours (79 frames)
Objective: 20× (0.215 μm per pixel)

E2: Cells treated with 50 ng/ml vascular endothelial growth factor (VEGF) vs. cells treated with phosphate-buffered saline (control)
Group 1: 4 sequences, 1646 cells
Group 2: 4 sequences, 1701 cells
Duration: 6 hours (73 frames)
Objective: 10× (1.02 μm per pixel)

E3: Cells treated with the EHT1864 inhibitor of Rac activity (EHT1864) vs. cells treated with dimethyl sulfoxide (control)
Group 1: 2 sequences, 1700 cells
Group 2: 3 sequences, 1520 cells
Duration: 19 hours (77 frames)
Objective: 10× (1.02 μm per pixel)

E4: Cells transfected with Nrp siRNA (siNrp) vs. cells transfected with non targeting siRNA (control)
Group 1: 3 sequences, 1221 cells
Group 2: 3 sequences, 1272 cells
Duration: 24.5 hours (50 frames)
Objective: 10× (1.02 μm per pixel)

E5: Cells transfected with vascular endothelial growth factor receptor 2 siRNA (siVEGFR2) vs. cells transfected with non targeting siRNA (control)
Group 1: 3 sequences, 1132 cells
Group 2: 3 sequences, 1390 cells
Duration: 22.5 hours (46 frames)
Objective: 10× (1.02 μm per pixel)

E6: Cells transfected with VE-cadherin-GFP adenovirus (VEcadGFP) vs. cells transfected with GFP adenovirus (control)
Group 1: 3 sequences, 387 cells
Group 2: 3 sequences, 397 cells
Duration: 26 hours (157 frames)
Objective: 20× (0.215 μm per pixel)

Appendix B
Author's Publications

B.1 Publications Related to the Thesis

B.1.1 Impacted Journal Papers

Cao, J., Ehling, M., März, S., Seebach, J., Tarbashevich, K., **Sixta, T.**, Pitulescu, M.E., Werner, A.C., Flach, B., Montanez, E., Raz, E., Adams, R.H., Schnittler, H.: Polarized actin and ve-cadherin dynamics regulate junctional remodelling and cell migration during sprouting angiogenesis. Nature Communications 8(1), 2210–2230 (dec 2017), https://doi.org/10.1038/s41467-017-02373-8
Authorship: [7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%–7.69%]
Number of citations: 2

- Kim, J., Cooper, J.A., Yap, A.: Septins regulate junctional integrity of endothelial monolayers. Molecular Biology of the Cell 29(14), 1693–1703 (2018), https://doi.org/10.1091/mbc.E18-02-0136, pMID:29771630

- Neto, F., Klaus-Bergmann, A., Ong, Y.T., Alt, S., Vion, A.C., Szymborska, A., Carvalho, J.R., Hollfinger, I., Bartels-Klein, E., Franco, C.A., Potente, M., Gerhardt, H.: Yap and taz regulate adherens junction dynamics and endothelial cell distribution during vascular development. eLife 7, e31037 (feb 2018), https://doi.org/10.7554/eLife.31037

B.1.2 Conference Papers Ranked as A in CORE

Sixta, T., Flach, B.: Multiple object segmentation and tracking by bayes risk minimization. In: Ourselin, S., Joskowicz, L. (eds.) Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016. pp. 607–615. Springer International Publishing AG, Gewerbestrasse 11, CH-6330 Cham (ZG), Switzerland (oct 2016)
Authorship: [50%–50%]
Number of citations: 0

B.1.3 Conference Papers Excerpted by WoS

Flach, B., **Sixta, T.**: Unsupervised (parameter) learning for mrfs on bipartite graphs. In: Burghardt, T., Mayol-Cuevas, W., Mirmehdi, M. (eds.) Proceedings of the British Machine Vision Conference. pp. 72.1–72.11. BMVA, Imaging Science, Stopford Building, University of Manchester, Oxford, United Kingdom (September 2013)
Authorship: [50%–50%]
Number of citations: 1

- Marinoni, A., Gamba, P.: Unsupervised data driven feature extraction by means of mutual information maximization. IEEE Transactions on Computational Imaging 3(2), 243–253 (2017)

B.2 Other Publications

B.2.1 Conference Papers not Excerpted by ISI or Scopus

Sixta, T.: Star convex object detection by the infinite shape mixture model. In: Kropatsch, W.G., Ramachandran, G., Torres, F. (eds.) CVWW 2013: Proceedings of the 18th Computer Vision Winter Workshop. pp. 2–8. Vienna University of Technology, Karlsplatz 13, Vienna, Austria (February 2013)
Authorship: [100%]
Number of citations: 0