Dissertation

submitted to the
Combined Faculties for the Natural Sciences
and for Mathematics
of the Ruperto-Carola University of Heidelberg, Germany
for the degree of
Doctor of Natural Sciences

Put forward by
M.Sc. Luca Fiaschi
born in Prato (Italy)

Oral examination: 20.11.2013

Learning Based Biological Image Analysis

Luca Fiaschi

05.09.2013

Referees: Prof. Dr. Fred A. Hamprecht
Prof. Dr. Ulrich Schwarz

The work associated with this thesis has been carried out under the auspices of Prof. Dr. Fred A. Hamprecht at the Heidelberg Collaboratory for Image Processing (HCI) between October 2010 and October 2013.

Contact: luca.fiaschi@gmail.com

Acknowledgements

If I had to pay back with real money to all people which supported me writing this Ph.D. thesis, I would be a big bankrupt. This section can just try to name some of them, unfortunately omitting the many others who have been also important in the past three years.

First of all I would like to thank Prof. Dr. Fred A. Hamprecht for his supervision. His support went far over being just a scientific father - he even tried to bribe me to make me quit smoking. He did not succeed in that, but his criticism and example made me became a much better scientist in the meanwhile.

In these three years I was fortunate to work together with Dr. Xinghua Lou. He is the mastermind and the main author of my first project - the quality control. More than this, the entire research developed in this thesis is much inspired to his work and to the enlightening discussions we had together.

For almost two years I had the honour to share my office with Dr. Ferran Diego. If I had any scientific or personal question he would just be there with the ready answer. Foremost, I thank him for his humanity and for travelling with me across the US. We had a wild time together.

I thank Dr. Bernhard X. Kausler for being just Bernhard, one of the smartest persons I ever met. He would not easily let you fit in any category but question your assumptions and spur you to be original. We travelled together Israel and Jordan and it was a very great time. He also translated the abstract of this thesis to German.

I thank Christoph Strähle and Swen Wanner for all the discussions we had in front of good German beer. During these moments, we used to speculate really fantastic algorithms that will revolutionize computer science. Yet, the morning after we would usually forgot what we talked about and go to work with a big headache. I also still owe them a lot money in rolling tobacco.

I thank Dr. Ullrich Köthe for teaching me all I know of C++ and for the clever discussion on the drafts of the articles. He has often been a living encyclopedia for any kind of subject.

I thank Dr. Anna Kreshuk for her patience sitting and debugging together. She first introduced me to the world of Python. I also thank her for all the wise advices she gave me and for checking my bad English spelling of most of the talks I gave.

I thank Thorben Kröger for his friendship and his clever and witty observations. He also was my first reference in any problem of GUI programming and 3D graphics.

I thank Konstantin Gregor for being the best student assistant I could hope for. His talent and passion made the working hours lighter and more fun. He also courageously took on his shoulders the big effort of annotating the datasets used in the experiments.

Along with those already mentioned, I would like to thank all the other outstanding people that I met in my group: Buote Xu, Svenja Reith, Gregor Urban, Martin Lindner, Joachim Schleicher, Robert Walecki, Philipp Hanslovsky, Kemal Eren, Oliver Petra, Rahul Nair, Dr. Christoph Sommer, Martin Schiegg, Dr. Chong Zhang and Dr. Melih Kandemir. I especially thank Dr. Björn Andres and Thorsten Beier for introducing me to the PUA world and for the discussions about probabilistic graphical models. In addition, I would like to thank Barbara Werner and Evelyn Wilhelm for their support with the German language and with the University administration. I will eventually miss all these people together with the Tuesday mornings of Journal-Club.

During the last three years I had the great opportunity to collaborate with some outstanding life scientists. I am very grateful to Dr. Marta Zlatic and Dr. Bruno Afonso for introducing me to the challenging problem of tracking Drosophila larvae. It has been especially fun to work with those little white guys. I am sure, they will keep showing up in my dreams (or nightmares) for a long time. I thank Dr. Jurgen Raymann for providing the dataset used in the first chapter experiments and I also thank Dr. Ana-Martin Villalba for supporting my Ph.D student application.

I thank the Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences (HGS) for three years of financial support. I would like to mention especially Dr. Michael Winckler and Oktavia Klassen for their administrative support.

I thank Darya Trofimova for offering me her friendship. It was really a pleasure to travel to conferences together. She took a great effort in correcting all the many spelling mistakes in the first draft of this thesis. Moreover, I thank my girlfriend Benedetta Regonesi, who made even greater effort in bearing me all these years. I always want her to read trough my work to make sure that what I wrote does make a bit of sense.

Finally, I would like to thank my family: Patrizia Baroncelli, Martina Fiaschi and Andrea Fiaschi for their loving support.

Sources and Figures

The material presented in this work draws on the research conducted by the author of this thesis and his collaborators. In particular, the material presented in Chapter 2 is published in *Lou et al.* (2012) which is co-authored by the author of this thesis. Dr. Xinghua Lou carried out the principal experiments which are presented in that chapter and is the main author of the article manuscript. The author of this thesis contributed with the prototyping of the algorithm and prepared the gold standard for the evaluation. Dr. Ullrich Köthe and Prof. Dr. Fred A. Hamprecht contributed with intellectual input and corrections on the manuscript. A third-party figure is used in that Chapter: Fig. 2.6 *Reymann et al.* (2009). Dr. Jurgen Reymann provided the dataset presented in the experiments of Chapter 2.

A part of the research material presented in the following chapters is published in *Fiaschi et al.* (2012, 2013). The author of this thesis wrote the principal code and conducted the principal experiments presented in those articles, with the input and the support from the all other authors of the manuscripts.

Zusammenfassung

Die aktuelle Forschung in Biologie und Medizin ist stark von den Fortschritten in der Entwicklung rechnergestützter Methoden abhängig. Die Explosion der Datenmenge in der Mikroskopie und das wachsende Interesse von Lebenswisschenschaftlern an immer komplexeren und subtileren Wechselwirkungen regen die Suche nach innovativen, rechnergestützten Lösungen für herausfordernde und praxisnahe Anwendungen an. Erweiterungen und Neuformulierungen generischer und flexibler Methoden basierend auf Lernverfahren bzw. probabilistischer Inferenz sind notwendig um die großen Vielfalt an produzierten Daten handhaben zu können und um ein ständiges Reimplementieren und ausuferndes Parameter-Tuning zu vermeiden. Diese Arbeit nutzt neueste Methoden des maschinellen Lernens basierend auf strukturierten probabilistischen Modellen und schwach-überwachtem Lernen um vier neue Ansätze in den Bereichen der Mikroskopie-Bildgebung und dem Multiple-Object Tracking zu präsentieren.

Kapitel 2 führt ein Framework für schwach-überwachtes Lernen ein um das Problem der Defektdetektion in Bildern aus großen Mikroskopie-Datenbanken zu lösen. Die vorliegende Arbeit demonstriert dabei genaue Vorhersagen mit nur geringem Annotationsaufwand von Seiten der Nutzer. Kapitel 3 präsentiert einen auf lokalen strukturierten Prädiktoren basierenden Lernansatz um in Bildern überlappende Objekte zu zählen. Dieses Problem findet vielfältige Anwendung in der Hochdurchsatz-Mikroskopie wie z.B. das Zählen von Zellen in der Toxizitätsanalyse von Medikamenten. Kapitel 4 entwickelt ein deterministisches graphisches Modell um zeitliche Konsistenz beim Zählen von Objekten in Videosequenzen zu erzwingen. Dieses Kapitel zeigt dass globale (zeitliche und räumliche) strukturierte Inferenz konsistent besser ist als lokale (ausschließlich räumliche) Vorhersagen. Die in Kapitel 4 entwickelte Methode wird in einem neuartigen Tracking Algorithmus verwendet, der in Kapitel 5 eingeführt wird. Dieses Kapitel nimmt sich—zum ersten Mal in der Literatur—dem schwierigen Problem an, überlappende, lichtdurchlässige und ununterscheidbare Objekte zu tracken. Die Behandlung von sich gegenseitig verdeckenden Objekten ist als ein neuartiges strukturiertes Inferenzproblem basierend auf der Mimimierung einer konvexen Multi-Commodity Flow Energie formuliert. Die optimalen Gewichte der Energieterme werden unter partieller Anleitung durch den Anwender mit Hilfe von latenten Variablen gelernt. Zur Unterstützung von Verhaltensbiologen wenden wir die Methoden auf das Problem an eine Ansammlung von interagierenden Drosophila Larven zu tracken.

As soon as an Analytical Engine exists, it will necessarily guide the future course of the science.

— *(***Charles Babbage 1864)**

Abstract

The fate of contemporary scientific research in biology and medicine is bound to the advancements in computational methods. The unprecedented *data explosion* in microscopy and the crescent interest of life scientists in studying more complex and more subtle interactions stimulate the research for innovative computational solutions on challenging real world applications. Extensions and novel formulations of generic and flexible methods based on **learning/inference** are necessary to cope with the large variety of the produced data and to avoid continuous reimplementation and heavy parameter tuning. This thesis exploits cutting edge machine learning methods based on structured probabilistic models and weakly supervised learning to provide four novel solutions in the areas of **large-scale microscopic imaging** and **multiple objects tracking**.

Chapter 2 introduces a weakly supervised learning framework to tackle the problem of detecting defect images while mining massive microscopic imagery databases. This thesis demonstrates accurate prediction with low user annotation effort. Chapter 3 presents a learning approach for counting overlapping objects in images based on local structured predictors. This problem has numerous applications in high throughput microscopy screening such as cells counting for drug toxicity assays. Chapter 4 develops a deterministic graphical model to impose temporal consistency in objects counts when dealing with a video sequence. This Chapter shows that global (temporal and spatial) structural inference consistently improves over local (only spatial) predictions. The method developed in Chapter 4 is used in a novel downstream tracking algorithm which is introduced in Chapter 5. This Chapter tackles, for the first time, the difficult problem of tracking heavily overlapping, translucent and indistinguishable objects. The mutual occlusion event handling of such objects is formulated as a novel structured inference problem based on the minimization of a convex multi-commodity flow energy. The optimal weights of the energy terms are learned with partial user supervision using structured learning with latent variables.To support behavioral biologists, we apply this method to the problem of tracking a community of interacting Drosophila larvae.

Contents

Chapter 1

Introduction

Microscopy has deeply contributed to important scientific discoveries in many disciplines and foremost in biology. Yet it is arguable that until recent years this technique has been mostly exploited to provide *qualitative* support to other experimental methods. Nowadays, we observe a large increase in the variety and quality of automatic microscopes some of which are able to produce huge datasets at extremely high resolutions. Modern microscopy opens the unprecedented opportunity to extract large-scale *quantitative* information by mining massive databases of images. Indeed, the collection and analysis of huge datasets is the general trend of most contemporary scientific research. This *flow of data* has given to some authors the feeling of facing a turning point of the scientific paradigm: from model/simulation based science towards data driven research (i.e. the fourth paradigm, *Hey et al.* (2009)). The impact of computational techniques has been pointed out as the driving force of this revolution (cf. *Myers* (2012); *Swedlow et al.* (2009); *Peng* (2008) for a discussion of image analysis in biology). As the opportunity for advanced data analysis is fastly increasing so are doing the expectations for more sophisticated algorithms. Indeed, data analysis and in particular image analysis often represents the bottleneck of ambitious research projects. Extracting quantitative information from biological images poses several challenges such as those of dealing with low signal to noise ratio, high data throughput and task sophistication. Behind these difficulties we should however not forget the opportunities: on the one hand to push forward the frontier of computational scientific methods, on the other hand, to open the door to important discoveries in life sciences. Honestly, the future could not look more exciting!

In the first part of this chapter we discuss in greater detail the challenges and the opportunities offered by biological image analysis. These opportunities motivate the focus of the work presented in this thesis - learning based methods for biological image analysis. Then, we review the trends in computational methods which are influential for our work. Finally, we present the structure of this thesis and overview the contributions introduced by this work.

1.1 On the challenges and opportunities of biological image analysis

The overwhelming challenge that needs to be addressed when studying living systems is *structural complexity* (*Adami*, 2002). This type of complexity eludes a precise mathematical formalization but it is easy to appreciate when looking at living organisms. Cells appear as rich chemical environments with a high level of spatial organization. Neurons exchange electrical and chemical signals in highly interconnected networks that end up defining actions for the entire organism. Even more surprisingly, patterns of collective behaviors, such as social interactions, emerge from the actions of individual animals (*Sumpter*, 2006).

Capturing the patterns in living systems requires a broad view over the spatial organization and the biophysical composition of complex systems at multiple scales: from nano to macro. Such a landscape exceeds the scope of single experiments and must be addressed by multiple experimental techniques. Ambitious goals can only be reached by large scale scientific projects that automatically extract data from several experimental methods and correlate the information of all sources. An example is the *Fly-Olympiad Project*[1] (*Branson et al.*, 2009; *Simon and Dickinson*, 2010) that aims elucidating the genetic and physiological basis of *Drosophila*'s behavior by integrating light microscopy (*Fly-Light*), electron microscopy (*Fly-EM*), genetic screenings and behavioral essays in a single massive database. Another example is the *Open-Connectome Project* [2] that coordinates a huge dataset of neuron images acquired from the nano scale (using electron microscopy) to the macro scale (using magnetic resonance imaging).

Computational science, and in particular *bioimage informatics* (i.e. the processing and analysis of images acquired with microscopes *Peng* (2008)), appears to be a central element in such large scientific projects. This is due to two main requirements that need to be fulfilled to support large scale biological science: *automation* and *repeatability*.

Automation is needed in order to address the size of the data generated with the newer microscopes. For example, as illustrated in Fig. 1.1, a modern high throughput microscope can produce 10^6 images in a single experiment, compared to the few hundred images produced by a traditional light microscope. Even more surprisingly, by using electron microscopy the imaging of a single Drosophila brain will produce a dataset of 150 terabytes, as estimated in *Myers* (2012). In addition, a substantial contribution to the increased throughput comes from techniques that produce multiple huge volumetric images, such as time resolved imaging (e.g. light sheet microscopy *Keller et al.* (2008)) or multiple channel imaging (e.g. array tomography *Micheva and Smith* (2007) and "clarity" *Chung et al.* (2013)). Yet, it is important to notice that the *type* of data that is produced is often out of scope for direct human analysis. For example, extracting useful information from mass

[1] `http://www.janelia.org/team-project/fly-olympiad`

[2] `http://www.openconnectomeproject.org/`

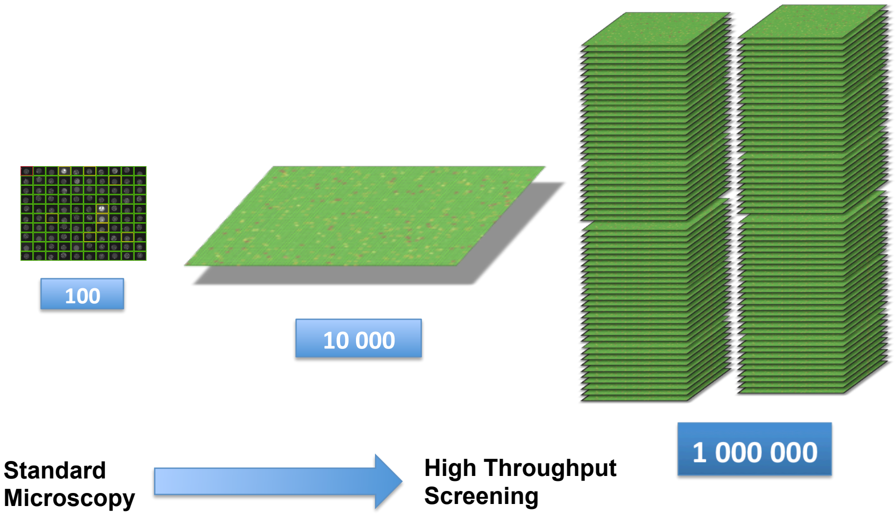


Figure 1.1: Illustration of the difference in data throughput between a traditional light microscope and an high-throughput microscope. In the figure, each colored square contains the image of a well plate. On the left, a traditional microscope can produce hundreds images of well plates per batch. In this regime, out-of-focus images (**yellow**) or images containing regional defects (**red**), such as debris contamination, are usually manually discarded. On the right, high throughput microscopy produces batches of $10^4 - 10^6$ images. Manual mining of defect images becomes impractical. An automated method for quality control of a large imagery database is presented in Chapter 2.

spectrometry imaging requires the inspection of hundreds of channels, whose effect cannot be even “visualized” without algorithmic pre-processing to reduce dimensionality (e.g. principal components analysis).

A second argument in favor of computational approaches is repeatability. Given the same input parameters and the same input data, an algorithm reproduces the same results (valid also for randomized algorithms provided the same seed as parameter). On the contrary, according to *Danuser* (2011), humans are subject to poor repeatability and are biased to overlook rare events. Even though committees of human experts are required to set the gold standard that is necessary to assess the algorithm performance, a single human expert can be outperformed in some tasks by algorithms. As argued in *Danuser* (2011), this can happen when the analysis requires the consideration of a large number of possible alternative solutions in a well-defined multidimensional space. For example this has been observed in protein location classification (*Murphy et al.*, 2003) or multiple particle tracking (*Matov et al.*, 2010). In the first case the algorithm performed better in the classification of visually subtle differences in location, as it could take advantage of a high dimensional feature space. In the second case humans were found to underestimate systematically fast movement events. In theory, an algorithm which performs tracking could take into account the position of all the particles in the sequence in order to establish a joint optimal assignment (this is limited by computational complexity as multiple-hypothesis-tracking *Reid* (1979) is an NP-hard combinatorial problem). In contrast, humans typically need to scroll back and forth through a sequence and inspect only a a limited number of particles at time.

Because large scale scientific projects have ambitious goals, the computational tasks that need to be addressed have become more and more challenging. Early bioimage informatics would deal mostly with image processing tasks, such as color enhancement, noise reduction, deconvolution and registration. These techniques produce an improvement of the images before further information is extracted. However, in order to deliver accurate quantitative measurements, scientific research demands today sophisticated image analysis solutions that require the extension of cutting edge methods in computer vision and machine learning (*Peng*, 2008; *Swedlow et al.*, 2009). For example, *Jain et al.* (2010) argue that the attempt to reconstruct the wiring diagram of the neurons in the brain (connectome reconstruction) is hindered by the lack of methods to accurately segment neuronal structures in microscopic images.

When analyzing biological images it is often required to deal with scenarios which are difficult to tackle using image analysis algorithms, such as low signal to noise ratio, low resolution and even ill posed problems. For example a group of objects which needs to be segmented, such as cells, could appear to overlap in the image and thus prevent the usage of simple techniques such as Otsu threshold (*Otsu*, 1975). To overcome these difficulties and enable the extraction of quantitative data from images, a careful modeling of the image content in order to take into account prior information about the scene is required. Depending on the task, the prior can be as intuitive as the observation that the objects under analysis do not escape the planar well plate, as exploited recently by *Fiaschi et al.* (2013) for counting heavily overlapping Drosophila larvae. Or it can be expressed in probabilistic terms and learned from data. In particular, machine learning techniques offer a powerful means to exploit prior information when present. For example, *Kausler et al.* (2012) propose a robust method for tracking a large number of divisible objects in presence of clutter. This approach integrates many terms which are learned from training data, such as cell appearance and mitosis probability, in a global probabilistic graphical model (PGM).

The evolution of techniques that integrate machine learning and computer vision may allow in the future to address fundamental questions in biology. For example, the tracking of thousands of cells in spatiotemporal volumes is required to elucidate embryogenesis (*Keller et al.*, 2008). In addition, image analysis may open the door to new field of studies. For example, the tracking of each individual in a large population of interacting animals is required to gain new insight into collective behavioral patterns. In particular, recent interesting work in this direction has been presented for adult Drosophila (*Branson et al.*, 2009), mice (*Branson and Belongie*, 2005) and ants (*Mersch et al.*, 2013; *Khan et al.*, 2006).

While pursuing ambitious research goals, particular attention needs to be given to the development of open-source software for bioimage informatics (*Cardona and Tomancak*, 2012). Indeed, well designed software enables also non image analysis practitioners to take full advantage of powerful computational techniques. It is interesting to note that, besides the large ecosystem of special purpose software, some interesting products have configured themselves as generic bioimaging tools and scientific workflow systems. The main advantage of scientific workflow systems is that generic components can be shared among several workflows without the need to be reimplemented. These components often include machine learning modules that give the flexibility to adapt to different datasets with the change of the training annotations (under similar experimental conditions). Often the implementation provided in these modules has been optimized to offer real time performance. This allows the user to interactively tune and refine the result (human-in-the-loop workflows). These software also often implement a plugin architecture which is continuously extended by active communities of developers.

Among the most used software, CELLPROFILER [3] and CELLCOGNITION[4] implement cell segmentation and classification modules that are largely used in cell phenotyping (*Carpenter et al.*, 2006; *Held et al.*, 2010). FIJI[5] and VAA3D[6] are used in the reconstruction and visualization of complex neuronal structures from large microscopy volumes (*Schindelin et al.*, 2012; *Peng et al.*, 2010). ILASTIK (*Sommer et al.*, 2011) provides a generic machine learning module for interactive pixelwise classification. It has been successfully applied to cell classification (*Sommer et al.*, 2012), synapses detection (*Kreshuk et al.*, 2011) and neuron segmentation (*Andres et al.*, 2012).

1.2 Recent developments in computational methods

In this section we shall review the advanced computational methods that are relevant for this thesis. A precise presentation of the related literature in connection with the novelty proposed in this thesis is found at the beginning of each chapter. Here we describe the most influential recent work with regard to three main areas of computer science: *machine learning*, *computer vision* and *optimization*. Nowadays these areas are strongly connected, most machine learning algorithms are indeed formulated as optimization problems while novel optimization techniques have been developed to address large scale machine learning. In addition, computer vision exploits both machine learning and optimization as fundamental building blocks and in turn motivates interesting research directions in both others fields. As a famous example, Neural Network (NN) (*Rosenblatt*, 1958) learning can be formulated as the problem of optimizing a non-convex energy function. The developments in gradient descent optimization (backpropagation) have made it possible to train Convolutional Neural

[3] `http://www.cellprofiler.org/`
[4] `http://cellcognition.org/`
[5] `http://fiji.sc/Fiji`
[6] `http://www.vaa3d.org/`

Networks (CNNs) (*LeCun et al.*, 1998) with thousands of neurons and large CNNs have recently shown outstanding results in difficult computer vision tasks, such as large scale category recognition (*Krizhevsky et al.*, 2012).

1.2.1 Machine learning

Probability is at the basis of machine learning and PGMs offer a diagrammatic representation of probability distributions (*Kollar and Friedman*, 2009; *Jordan*, 2004). The PGM language has been first developed to offer an intuitive representation of the conditional independence properties of high dimensional probability distributions, and it has evolved into a powerful modeling language that has unified many machine learning approaches (*Bishop and Nasrabadi*, 2006). The characteristics that have popularized PGMs are threefold:

- It provides building blocks (networks of random variables) that can be used to design complex probabilistic models.
- It comes with a sound theoretical analysis as well as a large set of algorithms for inference and learning.
- It makes explicit properties of the probabilistic model such as the conditional dependency rules.

In practice, a large number of real world problems involve inferring the assignments to variables which are related in a complex structure of dependencies. Most of these cases can be conveniently represented as PGMs. In particular, PGMs are instances of structured models (*Bakir et al.*, 2007).

In recent years, substantial attention in machine learning has been given to structured learning. Structured learning is a meta-parameter optimization procedure for structured models (see *LeCun et al.* (2006); *Nowozin and Lampert* (2011) and references therein for an introduction). Correlations between random variables influence two aspects of the learning problem:

- The output of the model can be structured. For example when performing semantic image segmentation, the segmentation output is interpreted as a graph connecting latent random variables associated with each pixel or superpixel (e.g. *Lucchi et al.* (2012)).
- The loss functions that measure the quality of a prediction can be structured. For example, when measuring segmentation performance the Rand Index is a nonlocal quality measure which is influenced by the label inferred for all pixels in the image (*Kroeger et al.*, 2013).

Several algorithms for structured learning have been proposed in the context of PGMs (*Roller*, 2004), Restricted Boltzmann Machines (RBMs) (e.g. *Mnih et al.* (2012)), and support vector machines (SVMs) (*Tsochantaridis et al.*, 2006). All these algorithms pose structured learning as the maximization of a scalar function (minimization of the energy) that defines the compatibility between structured input and structured output (e.g. the regularized loss function or a convex upper bound on the loss, cf. *LeCun et al.* (2006)). Among these algorithms, structured support vector machine (SSVM) has found widespread application in computer vision (*Nowozin and Lampert*, 2011). The success of SSVM is mainly due to two reasons: first, it introduces a regularization term in the loss function which reduces overfitting and second, it exploits a training procedure based on cutting plane minimization of a convex upper bound on the empirical loss. This procedure exploits the training algorithm of standard SVMs using the same efficient quadratic solvers.

Another very active area of research has focused on kernel methods. Kernel methods have been proposed in order to create an implicit non-linear data representation (*Schölkopf and Smola*, 2002) by exploiting the properties of the inner product (*kernelization*). In particular, a significant performance improvement in supervised classification, clustering, and regression has been shown for the family of SVMs algorithms (*Smola and Schölkopf*, 2004). Also SSVMs can be kernelized similarly to ordinary SVMs and have found recent applications in computer vision (cf. *Nowozin and Lampert* (2011); *Blaschko and Lampert* (2008)). However, the usage of kernelized SSVMs is still limited due to the high computational demand of using a structured kernel (i.e. *joint kernels*) (cf. *Lucchi et al.* (2012)).

1.2.2 Computer Vision

The big leap that computer vision and image analysis have taken in the last few years banks on advancements in *modeling*, *parameters optimization* and *features extraction*.

The design of computer vision models has been strongly influenced by the PGM language of machine learning. One very popular example is the application of Markov random fields (MRFs) theory in computer vision (*Blake et al.*, 2011). In particular, segmentation, depth estimation, denoising, matching and tracking have been posed as large scale inference problems. MRFs represent *un-normalized* probability distributions in terms of factorizable energy functions and are one type of PGMs. The Gibbs relation is used to establish the correspondence between the energy states and the configurations of an underlying probabilistic model.

The core ideas which are used in computer vision are:

- Images are represented as nodes assigned to pixels or superpixels
- Latent variables are associated with the nodes and represent the state of the pixel (e.g. a discrete label)
- A joint probabilistic model relates latent variables and pixel values within a graph. Prior knowledge (e.g. neighboring pixels should have the same label) is inserted in the model by defining factors over subsets of latent variables.

In order to minimize the discrete energy function of a MRF and to allow inference in large networks, powerful optimization methods have been developed such as graph cuts, tree reweighted junction belief propagation, linear programming (LP) relaxations and dual decomposition, these are discussed in Sect. 1.2.3.

The energy function of MRFs typically depends on a large set of parameters that are often hand-tuned while evaluating the algorithm. Setting these parameters is frequently the critical step to obtain the best performance from the method. Therefore, the parameters must be often retuned when the model is applied to a new dataset. This can often be a particularly tedious procedure and motivates the attention given to structured learning which can produce optimal parameters from a training set of annotated images.

One apparent limitation of learning algorithms it that the image annotation process is often very time consuming. Large scale crowd-searching engines have been developed to exploit the collective annotations of *non-expert* human labelers. For example, the *Amazon Mechanical Turk*[7] allows collecting thousands of training examples in few hours at a relatively low cost, and *LabelMe* (*Russell et al.*, 2008) is an open-source annotation tool available online [8] or on mobile platforms. However, such approaches are not applicable to labeling tasks which require expert supervision, like biomedical image analysis, or for unpublished and confidential datasets. In these cases, structured learning from partial annotations (*Lou and Hamprecht*, 2012) has been proposed as a means of reducing the labeling effort. In this framework, variables that are not labelled by the user or that are not accessible to direct observation are treated as latent variables in a similar fashion to latent structured support vector machines (LSSVMs) (*Yu and Joachims*, 2009). Active learning strategies (e.g. *Culotta and McCallum* (2005); *Small and Roth* (2010)) have been also combined with structured learning. These boost learning rate by querying user annotations only for the most effective instances.

Feature extraction has also received great attention in recent years. Depending on the downstream tasks, the features can be either extracted densely from each pixel, pooled over a group of pixels (e.g. Bag of Features *Csurka et al.* (2004)) or computed at sparse local key points (local descriptors). Two popular classes of features are convolutions with non-linear filter banks (e.g. Structure Tensor Eigenvalues *Bigun and Granlund* (1987), Gabor Filters *Jain and Farrokhnia* (1991)) and gradient histogram features (e.g. SIFT *Lowe* (1999) and HOG *Dalal and Triggs* (2005)). The extracted descriptors are often very high dimensional which may require feature selection (*Guyon and Elisseeff*, 2003). In contrast to the latter approaches, which use hand crafted sets of features, CNNs (*LeCun et al.*, 1998) have been proposed as a way to learn feature representations from raw pixel values directly. They achieve the state of the art classification results in challenging biomedical benchmarks (*Ciresan et al.*, 2012, 2013).

Computer vision has borrowed successful approaches from machine learning, such as ensemble learning, to reduce the impact of high dimensionality of the descriptors (i.e. curse of dimensionality) and limit over-fitting. A versatile ensemble

[7] `https://www.mturk.com/mturk/`

[8] `http://labelme.csail.mit.edu/Release3.0/`

method, Random Forest (*Amit and Geman*, 1997; *Breiman*, 2001), combines sets of decision trees. It has been shown to produce fast and accurate predictions in high dimensional classification, regression and density estimation tasks (*Criminisi et al.*, 2011). Random Forests have some useful properties for computer vision tasks. First, because of their hierarchical structure they have a fast learning and testing time. Second, they can be easily parallelized (even on the GPU (*Sharp*, 2008)), as each tree is independent from the others during training and testing. Third, classification Random Forests are inherently a multi-class algorithm and thus the use of heuristics such as "one-vs-all" or "one-vs-one" is not needed. Fourth, they can be easily and robustly trained on-line (*Saffari et al.*, 2009) in order to adapt to changes in the data distribution. In addition, *Breiman* (2001) argues that Random Forests can handle a significant amount of noisy features and noisy labels thanks to feature bootstrapping and the bagging of the training examples. For all these reasons, Random Forests find numerous applications in computer vision. They are exploited for semantic image labeling (*Shotton et al.*, 2008), in image classification and retrieval (*Bosch et al.*, 2007; *Moosmann et al.*, 2007), in object detection and tracking systems (*Gall et al.*, 2011; *Godec et al.*, 2012) and as an internal component of commercial visual sensors like KINECT[9] (*Shotton et al.*, 2013).

1.2.3 Optimization

The optimization community has been driven by the interest in Computer Vision for MRF models. In particular there has been a strong research in binary problems (the variables can only assume 0-1 values) that finds application in foreground-background segmentation. Large MRFs with binary submodular energy can be solved to optimality thanks to a polynomial time algorithm (min-cut/max-flow) *Boykov et al.* (2001). The recently proposed QPBO algorithm (*Rother et al.*, 2007) solves approximately general binary quadratic energies. However, integer programming is in general an NP-hard problem. For non binary problems, algorithms based message passing schemes such as Belief Propagations (e.g. (*Wainwright et al.*, 2005)) have been successfully proposed to obtain approximate solutions. In particular graphs, such as fully connected MRFs, message passing (Mean Field approximation (*Krähenbühl and Koltun*, 2011)) can be implemented very efficiently through convolutions. An other particularly powerful techniques is convex relaxation (*Komodakis and Tziritas*, 2007) that reduce the integer problem to solving LP optimization. Recently, *Kappes et al.* (2013) propose an extensive comparison of techniques for MRF energy minimization. The authors observe that for small and moderate problem sizes, advanced integer linear programming methods using cutting-plane and branch-and- bound provide the exact optimal solution and tend to be faster than other approximate methods. However, on large scale problems or for complex high order energies LP relaxation provides approximate solutions that are close to the optimum, within the shortest runtime.

[9] `http://www.xbox.com/en-US/KINECT`

Convex optimization has also been intensively investigated as driven by the interest in machine learning (*Sra et al.*, 2011). Among the most significant contributions to machine learning we shall mention methods for the solution of large quadratic problems, which find application in training of SVMs (*Platt*, 1998; *Joachims*, 1999), cutting plane for structured learning (*Teo et al.*, 2010), and stochastic gradient descent for large-scale learning problems (*Bousquet and Bottou*, 2007). This research has been beneficial also to some non convex problems that can be decomposed into convex subproblems (*Yuille and Rangarajan*, 2003), such as Difference of Convex functions programming (DC) (*Tao and An*, 1997) that is used in training of LSSVMs *Yu and Joachims* (2009).

To conclude this overview, it is important to note that all the exciting research directions of the last decade could not have been pursued without the constant advance of high-performance computing. In particular, it is worth to remember the improvements given by parallelism. On the one side, this includes technical improvements on dedicated hardware such as GPUs or multiple CPUs architectures. On the other side, it comprises all algorithms that allow decomposing a large optimization problem into a set of smaller ones (e.g. dual decomposition *Komodakis et al.* (2007)).

1.3 Thesis overview

The research directions developed in this thesis have been driven by specific applications in the field of biological image analysis. Each of the four central chapters propose a novel learning algorithm to address open challenges in two prominent areas of research: *large scale imaging* and *multiple object tracking*. These two related areas have primary interest in contemporary research in the life sciences. Indeed, as detailed in section 1.1, large scale imaging is the general trend of modern scientific research while multiple object tracking finds numerous applications in cells and animals tracking.

The inspiring principle of our work is to provide the users with methods based on learning that can be trained with reduced effort. This motivates our research and our proposals in the field of learning from partial as well as weak annotations. As reviewed in section 1.2.1, these learning approaches aim to keep the flexibility of supervised methods to adapt to the data while reducing the user annotation cost.

In particular, the content of each chapter is as follows:

- Chapter 2: In large scale imaging, quantitative data is gathered by algorithms on collections of images which are never directly seen by a human expert. High quality becomes a critical factor in such experiments in order to avoid jeopardizing the downstream statistical analysis. In this chapter, we propose a novel weakly supervised defect detection algorithm for large scale automatic microscopy screenings. While object detection is traditionally posed as a supervised classification problem, in our experimental setting the collection of enough representative examples for the rare and highly variable defect class can become a daunting task. The main contribution of this chapter is to perform defect detection with weak annotations that are faster to acquire.

- Chapter 3: One of the most useful assays in large scale automatic screenings is cell counting, which has important applications in the assessment of toxicity of newly developed drugs. Counting methods based on individual cell segmentation are strongly biased when cells overlap due to under-segmentation. This chapter builds on a recent approach for counting overlapping objects in images which avoids the hard task of single instance segmentation (*Lempitsky and Zisserman*, 2010). Only minimal user annotations are required: a dot in the centre of each cell. The main contribution of this chapter is a novel algorithm based on structured labels Regression Random Forests which greatly simplify the original method while maintaining similar performance.

- Chapter 4: Multiple object tracking finds numerous practical applications in bioimage informatics. The widely used class of *tracking by model-evolution* methods requires initialization of the tracked object (seeding). This step is often performed fully manually or with some heuristics. This chapter proposes a method that can reliably detect isolated individuals and thus can be exploited for automatically seeding downstream tracking algorithms. In addition, it delivers the number of objects in *each* foreground connected component and therefore detects mutual occlusion events (clusters of objects). Besides these results, which are exploited for the tracking algorithm of Chapter 5, the main contribution of this chapter is a temporally consistent structured model for object counting (a graphical model with deterministic potentials).

- Chapter 5: Clarifying complex animal behavior such as social interaction requires tracking of each individual in a large population. However, this is a particularly challenging problem when local features do not allow distinguishing between each target. This chapter proposes a novel structured learning algorithm for tracking overlapping translucent and indistinguishable objects. Our main contribution is an explicit modeling of the mutual occlusion dynamics formulated as a convex multicommodity flow problem. Unlike other tracking methods which require hand tuning, our approach finds its optimal parameters from *partial* user annotations.

All chapters are self-contained and organized following a similar scheme. First, the introduction of each chapter poses our contribution in the general context of the thesis with respect to the other chapters. Second, the relevant work for the problem is reviewed and the importance of our contribution is motivated. Third, the central sections introduce our model and demonstrate the validity of our approach with extensive experiments. Finally, we conclude each chapter by discussing the contributions of our method together with its assumptions and limitations.

A final discussion that summarises the presented material, highlight our main contributions and outlines future research directions is provided in Chapter 6.

Chapter 2

Quality Control of Microscopy Images

As argued in Chapter 1, a trait of contemporary scientific research in biology is the extensive usage of large-scale automatic microscopy. In this scenario, important information will be discovered by automatically analyzing the resulting image databases. However, an important prerequisite for an accurate analysis is a robust quality control of the images. Defect images can indeed invalidate the downstream statistical analysis or even mask out rare events such as rare cells phenotypes.

This chapter presents a weakly supervised learning framework to detect defects in high content screening (HCS). Unlike fully supervised object detection methods our proposal is to cast the problem as a novelty detection task based on one-class support vector machine (OCSVM) learning. The advantage of our approach is to avoid the tedious task of mining the dataset for defect images. Our training procedure requires only training examples for images of the abundant normal class. Our framework resembles a cascade of classifiers with feature and similarity measure designed for detecting different defect classes (i.e. global and regional). The latter characteristic, in combination with a fast runtime, could support the microscopic machinery in refining data acquisition in place.

This chapter is organized as follows: in section 2.1 we review the relevant literature on microscopy imaging quality control and outliers detection. In section 2.2 and 2.3 we discuss the distinct characteristics of the most common defects in HCS microscopic images and propose adequate similarity measures (features and kernels) at the different stages of a classifiers cascade. As reported in section 2.4, we evaluated this framework on a HCS database and obtained a 96.9% F-score for the important normal class. The contamination of the dataset with defects from the initial 10% is reduced to only 0.3%, providing an high quality dataset to downstream assays such as cell counting that is discussed in Chapter 3. The chapter is closed with a final discussion on the method in section 2.5.

2.1 Related work and contributions

High quality science depends more and more on high quality data. Indeed, large scientific projects, such as the Fly-Olimpiad[1] project, are acquiring massive databases in fully automated experiments. Quality control for large scale biomedical databases have been already address in the context of microarray *Shi et al.* (2006) while few attention has been given to imaging data. However, the key role that imaging plays in biomedical research (*Megason and Fraser*, 2007) have risen urgent demand for quality control of imagery databases (*Pepperkok and Ellenberg*, 2006). Most of the existing methods are based on manual inspection or semi-automated processing (*Goode et al.*, 2008; *Bray et al.*, 2012) and are not suited to address a large flow-of-data. Nowadays, the output of HCS (*Echeverri and Perrimon*, 2006) microscopy can reach millions of images per experiment and the newest microscopes (such as those used in connectomics (*Kaynig et al.*, 2008)) can produce 3D volumetric images which further increase the amount of data.

In this chapter we propose a scalable method for quality control in large scale HCS imaging experiments. Imaging biological data is a very sensitive process and defects can occur in multiple places (*Goode et al.*, 2008; *Bray et al.*, 2012). First, during sample preparation such as the contamination from debris or errors in the staining. Second, during image acquisition such as out-of-focus images or images with unheaven contrast and illumination. Fig. 2.1 shows examples of typical defects compared to normal images (Fig. 2.1 A and E). Defects are difficult to retrieve (minority class) and show a high variability in appearance. They can occur at the full image scale due to out-of-focus (Fig. 2.1 B, D and F), or only in confined regions within the image due to contamination or wrong staining (Fig. 2.1 C, G and H). Even though a quantitative analysis of the impact of defect images is beyond the scope of this chapter, we estimated that a typical HCS dataset, obtained with a state of the art microscope (*Reymann et al.*, 2009), can contain up to $5-10\%$ defect images. Given the sensitivity required in HCS studies (such as cell counting assays), this rate of contamination is non negligible and should be taken into account in any careful study because defects will jeopardize downstream analysis such as segmentation, registration and tracking.

Many challenges needs to be addressed for the quality control of HCS imaging databases. Firstly, supervised classification (based on support vector machine, random forest, etc. *Hastie et al.* (2001)) is very inefficient because the rareness of defect images makes it too time consuming to collect training examples. Indeed, the acquisition of a sensible training set for the defects can require manual inspection of a large part of the dataset. For example defect detection could be performed with a generic object detection algorithm (e.g *Viola and Jones* (2004)), but in practice this approach is very expensive in terms of annotation effort as it requires several training examples. Secondly, it is also difficult to characterize directly the statistical distribution of the defects because of the large variability in scale and appearance

[1] `http://www.janelia.org/team-project/fly-olympiad`

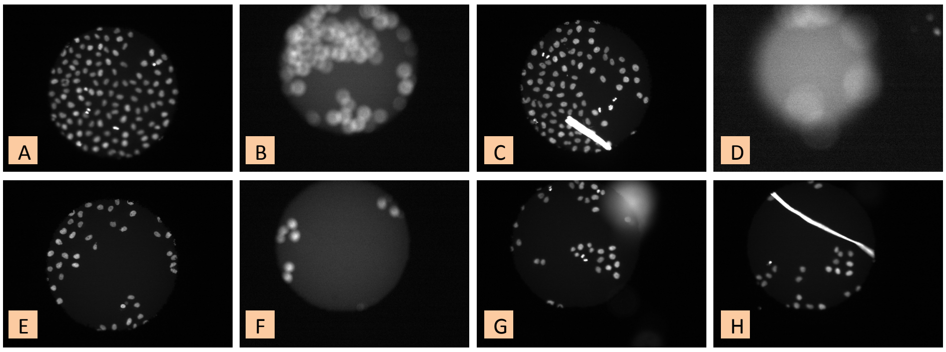


Figure 2.1: Examples of normal and defect images in the database acquired with a state of the art high-content screening microscope (*Reymann et al.*, 2009). In the table: (A) and (E) normal images; (B), (D) and (F) out-of-focus; (C), (G) and (H) debris.

(*Bray et al.*, 2012). Finally, the data throughput from modern microscopes makes manual inspection impracticable while automatic analysis requires algorithmic scalability and the support of parallel computing (*Goode et al.*, 2008). To tackle all these challenges we pursued two goals in the design of our framework: low labeling efforts and high scalability.

We cast the quality control task as an novelty detection problem (*Chandola et al.*, 2007) (i.e. defect images as anomalies) and chose to develop our framework based on the OCSVM (*Schölkopf et al.*, 2001; *Tax and Duin*, 1999). Briefly, in a novelty detection problem the training data is not polluted by outliers and we are interested in detecting anomalies in new observations. This is related to an outliers detection problem where the training data already contains outliers and the task is to fit the central mode of the distribution, ignoring the deviant observations. During the training phase of our method, the user provides labels for the normal images only. These samples are fed to a cascade of OCSVM classifiers that, in a projected space by kernalization, find the most compact "ball" to enclose the training data. The mathematical formulation of OCSVM is explained in detail in section 2.3.1. Test samples outside this ball (i.e. the decision boundary) will be classified as outliers. While a single OCSVM is, in theory, sufficient to detect distribution outliers, we propose to use a cascade of classifiers which builds a hierarchical image representation. Anomalous images are captured at multiple scales leading to two main advantages. First, running time is comparable with data acquisition time and second and the ability to discriminate between regional and global defects. Both properties are appealing because they can allow a feedback loop while acquiring the images. For example, an out-of focus detection could trigger a refocusing of the microscope without need to discard the data.

In summary, our major contribution is a one-class classifier cascade workflow which copes with various causes of anomalies given only labels for normal images.

This framework achieves a good scalability and accuracy while simplifying the training procedure for the user.

2.2 Defects in Microscopic Images: Global vs. Regional

We group common causes for image defects into two classes, depending on whether they affect the image *globally* or *locally*. A typical cause for global defect is out-of-focus imaging and typical examples of regional defects such as debris contamination (e.g. hair) (*Bray et al.*, 2012). We handle these two types of defects differently with appropriate features and similarity measure, which allows to predict three classes (normal, globally defect and regionally defect) even when training samples are provided for the normal class only. The advantage of keeping these classes separated is that when global defects (due to out of focus) are detected during image acquisition, they can be immediately corrected by retaking the image without need to discard the data.

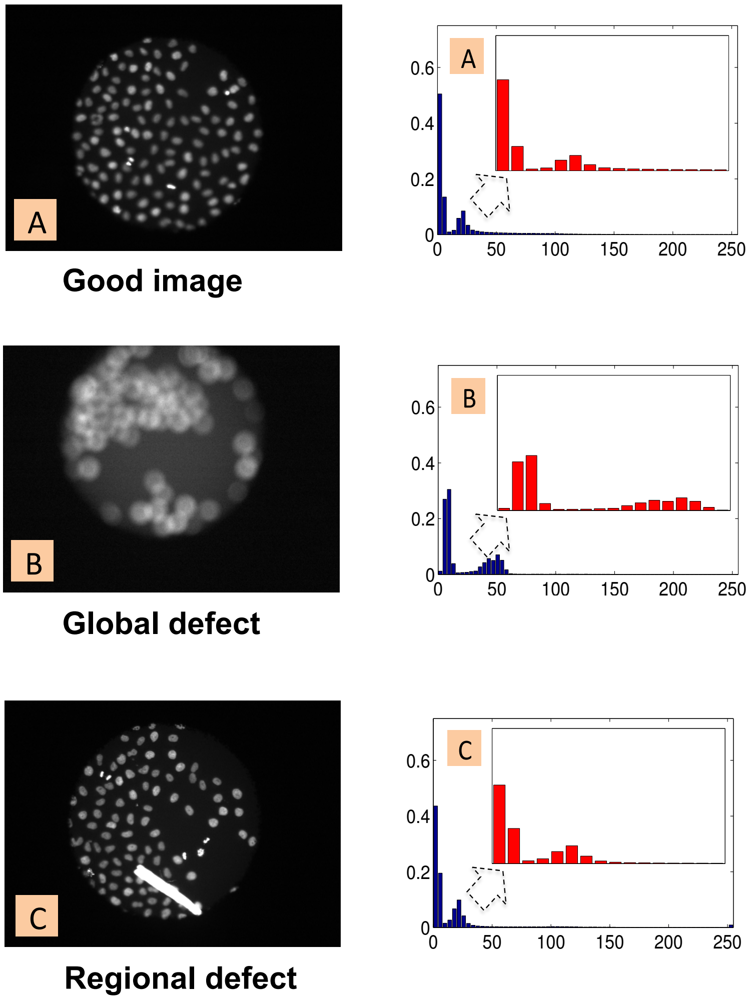
A characteristic of global defect is that they can be detected from the statistics drawn from the entire image. The formation of images is the convolution of the real light with the point spread function (PSF) (*Born and Wolf*, 1999). When out-of-focus occurs, the PSF becomes wider, and this can be seen from the intensity histogram of the gradient magnitude drawn from the entire image (e.g. Fig. 2.2 A vs. Fig. 2.2B, intuitively small texture edges with low magnitude are filtered out while strong edges are blurred).

The task becomes more difficult when regional defects occur, because they exhibit considerable variability in scale, position and shape. A global statistic is no longer informative, e.g. the histogram of the gradient magnitude in Fig. 2.2A (normal) is very similar to Fig. 2.2C (regional defect), and extracting information from fine regional details becomes necessary. In addition, regional defects show significant variability in appearance and scale, implying the requirement of more features to achieve a sufficient discriminative power.

2.3 Classification by One-Class SVM Cascade

The proposed quality classification framework is shown in Fig. 2.4 and consists of multiple stages which handle different defect classes. The first stage aims to filter out global defects by operating on the full image. The second and the third stage detect the defects at patch level. The second stage implements a coarse path classifier and the third one conveniently refines the results obtaining an overall speedup in the regional defects detection task.

The framework is similar to a cascade of OCSVM classifiers that use different kernel metrics (an earth mover's distance (EMD) kernel for stage one and two radial basis function (RBF) kernels for stage two and three, as depicted in Fig. 2.3) in order to maximise the tradeoff between speed and performance.



Good image

Global defect

Regional defect

Figure 2.2: Examples of the gradient magnitude histograms of normal and defect images. From row (A)-(C), left raw image and right corresponding histogram. The red histogram inside is the zoomed view showing the intensity range of interest (between 0 and 64).

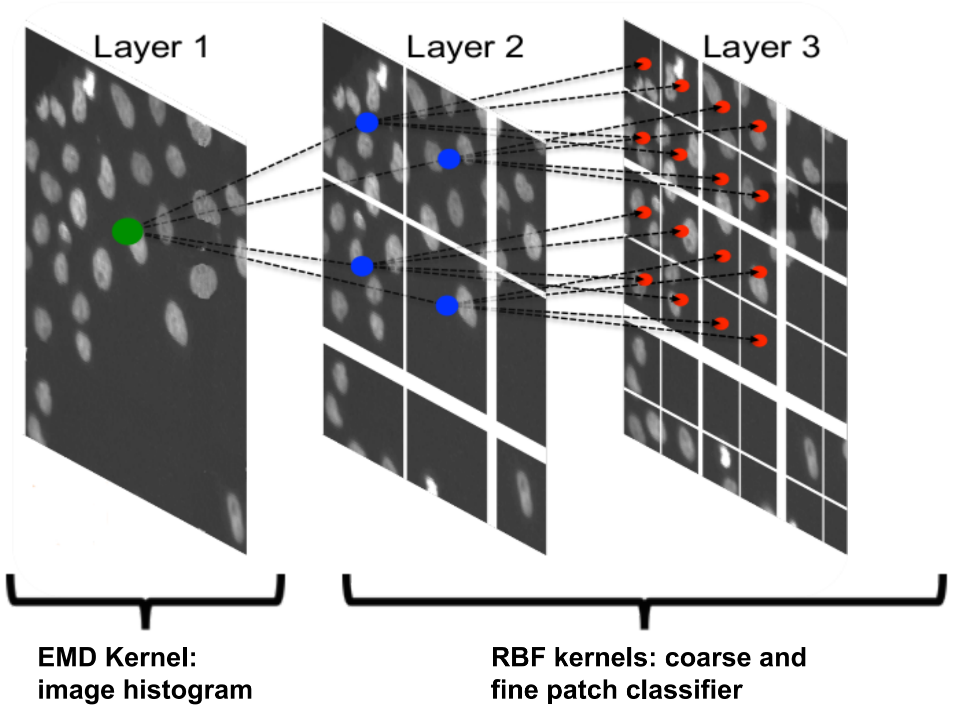


Figure 2.3: Illustration of the connection between the different layers of classification. The first layer uses a OCSVM with a EMD kernel to filter out out-of-focus images. Next layers implement a coarse to fine local outlier detection by ensembles of OCSVMs with RBF kernels.

2.3.1 One-class SVM

In the literature several outliers and novelty detection algorithms have been proposed. Among the most used are statistical models (*Hero*, 2006), distance measure (*Knorr and Ng*, 1998), density estimation (*Breunig et al.*, 2000) or space partition (*Liu et al.*, 2008a). Compared to these models the OCSVM (*Schölkopf et al.*, 2001; *Tax and Duin*, 1999) has the advantage of being easy to kernelize. Such property is often desirable when handling image data with an high variability in appearance and becomes even more important when handling histogram features which require a kernelized similarity measure to fully leverage a cross bin metric (e.g. earth mover's distance based (*Rubner et al.*, 1998)). For example, isolation forest (*Liu et al.*, 2008a) offers high scalability by exploiting an example of decision trees that can be trained in parallel, but are limited to the original feature space.

The *primal* form of the OCSVM problem is posed as the task of fitting a ball of minimum radius R to a set of training examples $\{\boldsymbol{h}_i\}$, $i = 1,...,N$ and $\boldsymbol{h}_i \in \mathbb{R}^M$. This ball is defined in a P dimensional feature space given by the mapping $\boldsymbol{\phi}(\boldsymbol{h}_i) : \mathbb{R}^M \mapsto \mathbb{R}^P$, and it is centred in $\boldsymbol{c} \in \mathbb{R}^{\mathbb{P}}$. In formulas:

$$\begin{aligned} \min_{\substack{R\in\mathbb{R}^+ \\ \xi\in\mathbb{R}^N_+ \\ \boldsymbol{c}\in\mathbb{R}^P}} \quad & R^2 + \frac{1}{N\nu}\sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\boldsymbol{\phi}(\boldsymbol{h}_i) - \boldsymbol{c}\| \leq R^2 + \xi_i, \ \forall i. \end{aligned} \tag{2.1}$$

The ν parameter, also known as the margin of the OCSVM, is proportional to the probability of finding a new observation outside the ball frontier. This frontier defines the decision boundary which separates inliers from outliers at test time. Kernelization is made explicit by the *dual* form of problem 2.1

$$\begin{aligned} \min_{\alpha\in\mathbb{R}^N} \quad & \sum_{i,j} \alpha_i\alpha_j K(\boldsymbol{h}_i, \boldsymbol{h}_j) - \sum_i \alpha_i K(\boldsymbol{h}_i, \boldsymbol{h}_i) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{1}{N\nu}, \ \forall i \\ & \sum_i \alpha_i = 1 \end{aligned} \tag{2.2}$$

Where $K(\boldsymbol{h}_i, \boldsymbol{h}_j) = \langle \boldsymbol{\phi}(\boldsymbol{h}_i), \boldsymbol{\phi}(\boldsymbol{h}_j) \rangle$ is the kernel matrix that represents the scalar product in the mapped feature space. The solution of the quadratic optimization problem 2.2 can be obtained efficiently with the SMO algorithm (*Platt*, 1998). The solution is a non-linear decision function in the original space of the form:

$$f(\mathbf{h}) = \operatorname{sgn}\left(R^2 - \sum_{i,j} \alpha_i\alpha_j K(\boldsymbol{h}_i, \boldsymbol{h}_j) + 2\sum_i \alpha_i K(\boldsymbol{h}_i, \boldsymbol{h}) - K(\boldsymbol{h}, \boldsymbol{h}) \right) \tag{2.3}$$

For completeness, we note that the OCSVM algorithm was independently formulated by *Tax and Duin* (1999) and *Schölkopf et al.* (2001). The two formulations can be shown to be equivalent up to a transformation of the kernel matrix *Schölkopf et al.* (2001).

2.3.2 Global Out-of-Focus Detection by Histogram Comparison

Out-of-focus (blur) detection has been recently studied in the context of natural images (*Liu et al.*, 2008b). In that study it is evidenced that out-of-focus mainly affects the high frequency spectrum of the gradient since small edges (texture) are washed away from the image. To leverage this observation we collect the histogram of the Gaussian Gradient Magnitude of the image at a small scale. We then compare each two histograms with a histogram kernel based on EMD. This kernel is used for the first OCSVM that detects out-of-focus images (stage one in Fig. 2.4). EMD (*Rubner et al.*, 1998), which was developed in the context of optimal transport theory as an optimization problem, represents the cost for transporting probability mass ("earth") from one distribution (i.e. normalized histogram) to the

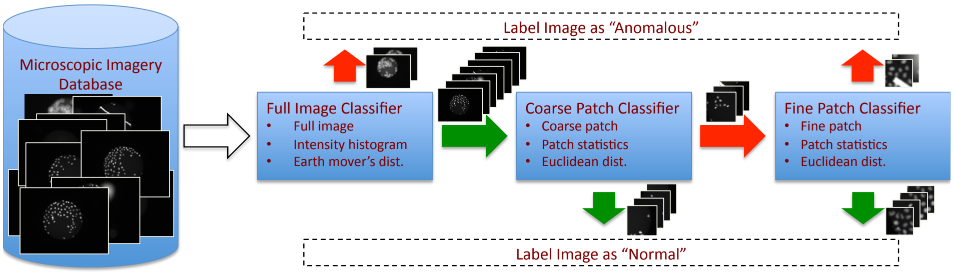


Figure 2.4: The proposed workflow is a OCSVM cascade. The microscopic imagery database is screened in three different stages that corresponds to different scales for the defects. Red and green arrows indicate the flow of detected defects and normal images (patches), respectively.

other. EMD based histogram comparison has proven superior to the Euclidean distance measure (*Rubner et al.*, 1998) since it is a cross bin metric; the latter is however computationally much cheaper. In fact, the computation of the EMD involve the solution of a LP optimization. Efficient implementations have been studied in (*Pele and Werman*, 2009). Formally, given two histograms ($\boldsymbol{h}_i$ and $\boldsymbol{h}_j$) normalized histograms the kernel for out-of-focus detection is:

$$K^{\text{EMD}}(\boldsymbol{h}_i, \boldsymbol{h}_j) = \exp(-\lambda^{\text{EMD}} EMD(\boldsymbol{h}_i, \boldsymbol{h}_j)) \tag{2.4}$$

The parameter λ^{EMD} adjusts the scale of the kernel response. In order to have a valid kernel for OCSVM the histogram must be L1 normalized (the sum of the histogram must be one) (*Pele and Werman*, 2009).

2.3.3 Regional Defect Detection from Patch Statistics

In Sect. 2.2 we have motivated the need for a multiple scale analysis. In particular regional defects cannot be detected from the full image statistics (histogram) and require inspection at finer scale such as dividing the images in smaller patches. There are two aspects which must be considered when moving to a finer scale. First, the increased complexity of the task must be considered: hundreds of patches may need to be extracted per image. In a datasets of thousand of images this yields to a new problem with million of patches. Second, regional defects can occur at any location and exhibit a high variability in appearance (c.f. Fig. 2.1), thus they may require a richer feature representation to be discriminated. We use two techniques for regional defect detection. Firstly, we draw basic statistics from low level texture, edge and intensity features and use RBF kernel for patch similarity measure. Secondly, we construct a coarse-to-fine procedure for speedup. These two stages are described in the following subsections.

Low Level Features and Patch Statistics

In order to compute a rich feature representation for the patches, we use a filter bank including texture (structure tensor), edge (gradient magnitude) and line (eigenvalue of the Hessian) detectors. The filter bank is computed at multiple scales and for each scale we collect the following statistics for each patch: mean, standard deviation and quantiles (10%, 50% and 90%). It is to note that such features can be computed quite efficiently. First, the filter bank requires only convolutions of the image. Second, the filter response statistics can be computed by a single swipe trough the images using statistics accumulators such the ones implemented in the BOOST library [2]. It is also important to note that patch statistics and in particular quantiles offer a compressed representation of the histogram of the patch which is more memory efficient. To compare patch statistics we exploit a RBF rather than the EMD: EMD is more expensive than Euclidean distance (for RBF kernel) by several orders of magnitude because it requires solving an LP optimization problem for each two query points.The explicit form of the used RBF kernel is:

$$K^{\mathrm{RBF}}(\tilde{\boldsymbol{h}}_i, \tilde{\boldsymbol{h}}_j) = \exp(-\lambda^{\mathrm{RBF}} ||\tilde{\boldsymbol{h}}_i - \tilde{\boldsymbol{h}}_j||_2^2) \tag{2.5}$$

where λ^{RBF} is a parameter and $\tilde{\boldsymbol{h}}_i$ is the i^{th} patch statistics vector.

On Feature Bagging and Classifier Ensembles

The patch statistics is used as input features to the OCSVM. Since the filter bank is computed at multiple scales this procedure results in an high dimensional feature vector. To reduce the noise introduced by correlated and less discriminative features (course of dimensionality) we exploit two strategies: feature bagging and classifier ensemble (*Lazarevic and Kumar*, 2005). Inspired by the algorithm of the Random Forest Classifier (*Breiman*, 2001), we sample multiple subsets of features (a bag of features) and train a OCSVM for each subset. The number of features in each bag is equal to the logarithm of the total number. In practice, important features become more influential in a lower dimensional space therefore we found out experimentally that averaging votes from the ensembles is more robust than a single OCSVM trained on all features.

We illustrate the benefits of using an ensemble of classifiers with a control experiment. We prepare a control set of 1000 randomly selected patches by taking 500 patches from the normal images and 500 patches from the defect images and we qualitatively compare in Fig. 2.5 the RBF kernel obtained by using all features and the average kernel obtained by using feature bagging. In an ideal situation (Fig. 2.5A) all the patches coming from the normal images will be similar to each other (high kernel response in white), while cross similarities between normal images and outlier patches will be not present (no kernel response black). Unlike in the supervised two class kernel case, in the context of OCSVM, the outliers

[2] `http://www.boost.org/doc/libs/1_54_0/doc/html/accumulators.html`

patches from defects images do not have to be similar to each other as they can be distributed in an arbitrary way outside the decision function and thus on average they may give a noisy response in the ideal kernel (note that the actual appearance of this region of the kernel is not influential for the task). As we can see by comparing Fig. 2.5B and Fig. 2.5C the contrast between normal and defect samples is strongly enhanced for the average kernel case (Fig. 2.5C) which will result in an increased discriminative power for the ensemble of classifiers.

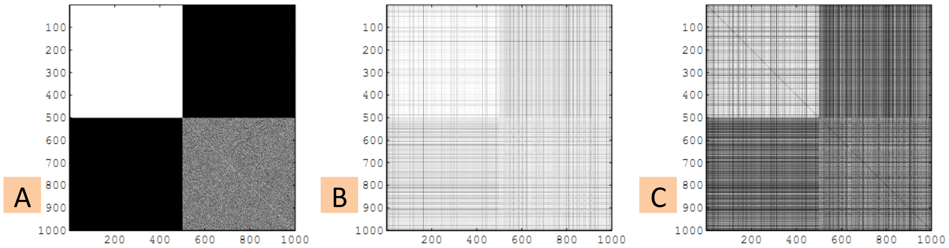


Figure 2.5: Kernel matrices for 500 normal and 500 defect samples: (A) ideal kernel; (B) kernel using all features; (C) average kernel from feature bagging. The average kernel from bagging has higher contrast between the inlier and outlier classes and it is qualitatively closer to the ideal kernel.

Two Steps Patch Filtering Procedure

To further speedup the proposed workflow we observe that a significant amount of image regions are easy to classify as normal ones (such as background, regions with sparse objects). Thus we divide the detection of regional defects in two stages which operate respectively at a coarse and a fine patch scale (stage two and three in Fig. 2.4). The purpose of the coarse patch classifier is to filter out the image regions which are easy to recognise as normal and thus reduce the workload of stage three which works on smaller patches (and therefore is more expensive). Similarly to what happen in the transition between global and regional defects (between stage one and two in Fig. 2.4), large patch size may average out small defect regions and produce a false normal patch. To prevent this, we made stage two more selective on determining normal images by setting a high ν value (cf. Eq. (2.1)) to the OCSVM (*Schölkopf et al.*, 2001).

2.4 Experiments

Our framework is evaluated on an image database for mammalian cell culture study acquired with state of the art microscope for HCS (*Reymann et al.*, 2009). The 9216-microwell cell array (in a 96×96 layout) yields to one image per well (9216 in total) as illustrated in (Fig. 2.6 and Fig. 2.7). The automated microscope took an overall imaging time of around 10 hours (roughly 4 seconds/image), this is comparable with the time required to run our workflow on the entire dataset.

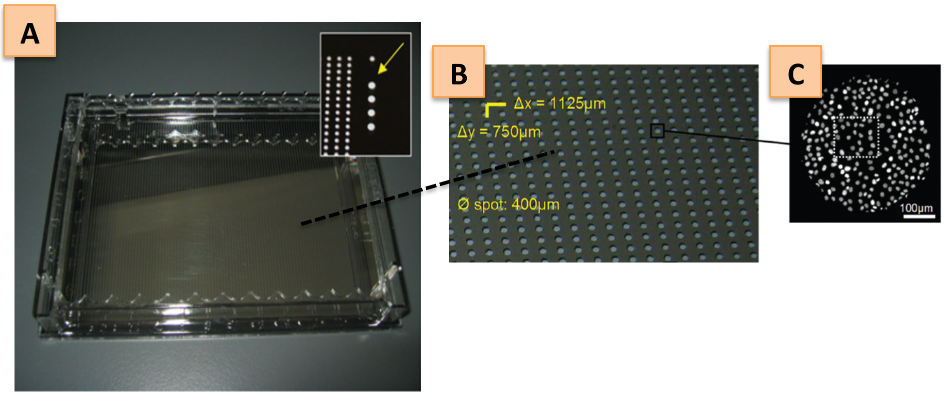


Figure 2.6: Image modified from *Reymann et al.* (2009). On the left, (A) 9216-microwell cell array for high content screening microscopy. On the right, (B) zoom inside the array and (C) on a single well.

2.4.1 Implementation details

The workflow is implemented in Matlab. For the quadratic solver needed in the OCSVM we use the SMO algorithm from `libqp`[3]. Filter banks computation is implemented using `VIGRA`[4], while we used the efficient code from *Pele and Werman* (2009) for computing the EMD. The entire code for our workflow is available upon request. During the training, it is optimal to select a representative set of normal images with different characteristics (e.g. cell density, illumination, etc.) such that the one-class classifier response (at all stages of the workflow) does not produces a biased classification. We also used an incremental training of the algorithm. We started with some training images and alternate training and prediction repeatedly while visually checking the performance of the algorithm on a small validation set. We made two rounds of incremental learning and eventually found 140 (out of 9216) training images. Since the size of the training set is statistically negligible we decided not to exclude it from the test set. This procedure is representative of a real world workflow scenario. Overall, the framework took roughly 2.5 hours to complete the prediction on the entire dataset on a 4 core (2.8G-Hz) machine; Training time is of the order of 5 minutes per stage.

2.4.2 Results

To establish the golden standard of our evaluation the entire dataset was labelled by a human expert. In our research we are interested in cell segmentation and counting, therefore the manual ground truth is generated accordingly. It is important to note that the amount of blur which is tolerated in the images can change with the scope of the analysis. For example it is difficult to extract the phenotype from

[3] `http://cmp.felk.cvut.cz/~xfrancv/libqp/html/`

[4] `https://github.com/ukoethe/vigra`

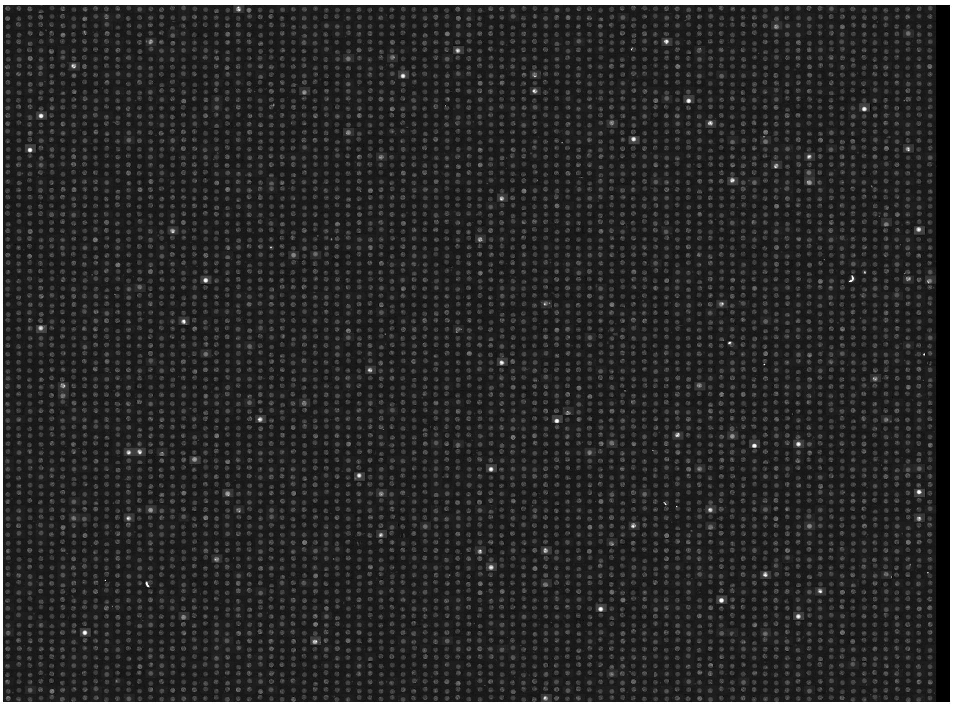


Figure 2.7: Mosaic of a subset of the images from the dataset used in the evaluation. Bright spots corresponds to clear out of focus well plates. Regional defects require local inspection of the image to be detected.

out-of-focus images, since this task could require an accurate classification of the cell texture. Otherwise it is still possible to use a heavy out-of-focus image for cell segmentation and counting. One of the advantages of an approach based on learning is that the same workflow can transferred more easily to a different task by changing the training images and selecting appropriate features.Table. 2.1 shows the overall detection accuracy by our framework, which is depicted as a confusion matrix (rows being the ground truth), and the per class precision/recall is given in Table 2.2. To summarize this result with a single number we note that the initial contamination of the dataset is 10% while after running this framework, it is reduced to 0.3%. During the analysis only 5.8% of good images are discarded.

	Normal	Out-of-Focus	Regional
Normal	7854	146	338
Out-of-Focus	1	426	28
Regional	19	47	357

Table 2.1: Confusion matrix for the classification task. The most confused class is false positive regional defects. The parameters were slightly biased during training since it is more costly to introduce defects among the normal images.

	Precision	Recall	F-score
Normal	0.997	0.942	0.969
Out-of-Focus	0.688	0.936	0.793
Regional	0.494	0.844	0.623

Table 2.2: Per class precision and recall.

Some examples of detected regional defects are shown in Fig. 2.8. Our framework shows good accuracy on detecting regional defects considering that they exhibit strong variability in size, shape, texture and considering there was no annotation of this class. Fig. 2.11 shows some errors by our framework. We notice that misdetection occurs when the regional defect is not sufficiently bright (left two images). In such cases the intermediate coarse patch classifier may mistakenly decide to consider the region as a normal region.

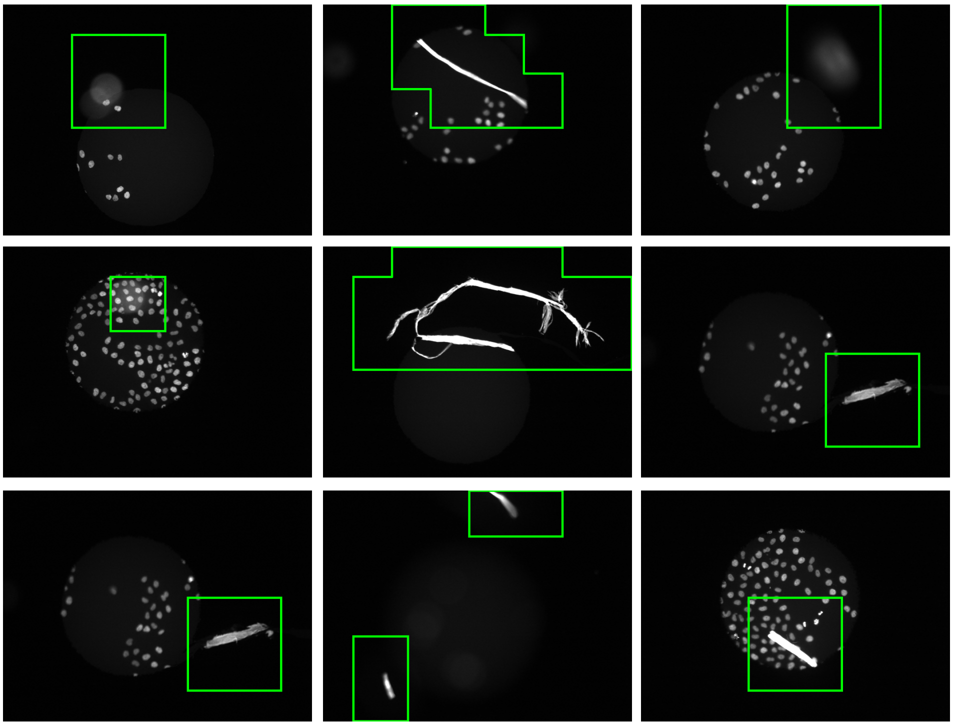


Figure 2.8: Examples of regional defects found by our framework. Regional defects have high variability in scale and appearance however our framework shows a good detection accuracy.

Some examples of detected global defects are shown in Fig. 2.9. In Fig. 2.10 it is shown the signed distance to the decision boundary of the out-of-focus classifier for each image in a 96 × 96 cell array layout. Higher red value indicates a more strong out-of-focus defect. The images are acquired by scanning the well plate row by row, the focusing machinery of the microscope is initialized at beginning

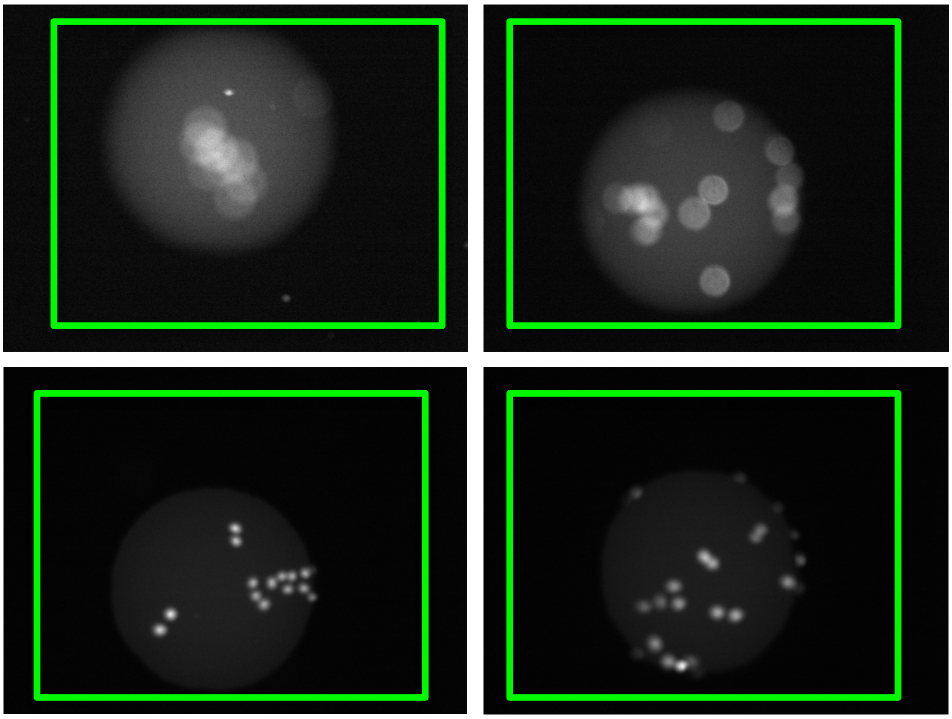


Figure 2.9: Examples of global defects (out-of-focus) found by our framework. The strength of the parameters ν and λ_{EMD} control the amount of blur which is tolerated by the algorithm. These parameters where set manually after visual inspection of a small validation set.

of each row for speedup. The evident stripe in the center of the array is due to a hardware error that caused the microscope to malfunction during the scanning of row 45 and 46 . The detection of such systematic errors can help the design of future experiments.

2.5 Discussion

Unlike natural images which are selected by the photographer to be visually appealing and free from technical errors, microscopic images present often defects due to the automatic acquisition process and the sample preparation. This Chapter prepares the way to further processing of HCS imagery datasets by presenting a framework for microscopic image quality control based on one class learning. In this chapter we study the characteristic of the typical defects which are found in a HCS imagery dataset acquired with a state of the art microscope. Based on our observations, we show that its possible to distinguish global and regional defects with a scalable cascade of one class classifiers and we propose appropriate features for distinguishing these classes. It is worth noticing that our method requires only

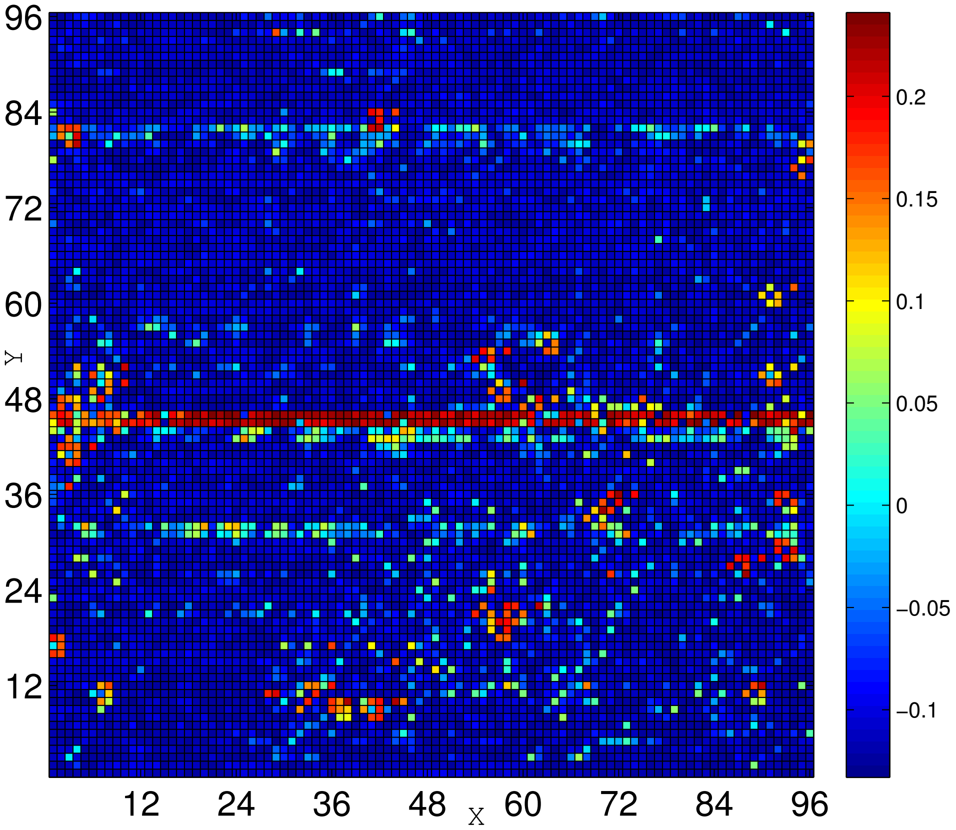


Figure 2.10: Location of out-of-focus images on 96 × 96 cell array layout. The color represents the signed distance to the classifier's decision boundary. Higher value signals a severe out-of-focus error. The prominent strip in the center indicates an error in the focusing system of the microscope during acquisition.

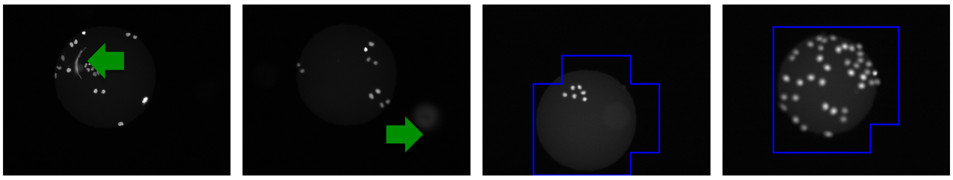


Figure 2.11: Examples of error in the detection of regional defects. Left two, two missed detections of regional defects with a weak intensity. Right two, two false positive detections where dark and slightly out of focus images are detected as regional defects.

training images for the normal class. In this imaging regime the preparation of a representative training set would be too time consuming for fully supervised multi-class learning as it would require mining a large part of the data. Our approach avoid this costly labelling effort.

In the future, multiple extensions of this research are possible. We showed that the running time of our framework is comparable with data acquisition time. This can be a practical benefit as it can allow to correct detected out-of-focus images while the microscope is scanning the well plate. In addition, given the generic structure of the classifier cascade, we would like to extend our framework to other imaging modalities. One realistic scenario is the detection of missing slices, a defect that often occurs in volumetric electron microscopy imaging.

Chapter 3

Learning to Count with Regression Forest and Structured Labels

Counting objects in images or video frames is important in many real-world applications including industrial inspection, cytometry, surveying and surveillance. For example, in HCS a common assay to asses the toxicity of a certain drug is to count the number of cells which survive after treatment. If the objects in the image are isolated and their appearance makes them distinct from the background, the simplest approach is to segment the image and count the number of foreground connected components. However, when multiple objects overlap this method provides a poor estimation because several instances can be merged in a single connected component (a cluster). This chapter investigates a recently proposed method for counting objects in static images (*density-counting Lempitsky and Zisserman* (2010)) that avoids the hard task of segmentation and detection. This approach seeks to count objects by integrating over an object density map that is regressed from local image features.

This chapter introduces a novel density-counting algorithm. Unlike previous work (*Lempitsky and Zisserman*, 2010)) where the prediction for each pixel is independent, we obtain the density map by averaging over structured, namely patch-wise, predictions. Using an ensemble of randomized regression trees that uses dense features as input, we obtain results that are on par with the state of the art methods but require only a fraction of the training time and lower implementation effort. Given the numerous possible applications of this algorithm, an open-source implementation is currently under development in the framework of `http://ilastik.org`. The software will provide a graphical user interface to faster training of the algorithm on different datasets.

3.1 Related work and contributions

Recent work has shown that object counting can be solved with equal or better accuracy *without* prior object detection or even segmentation (*Lempitsky and Zisserman*, 2010; *Ryan et al.*, 2009; *Chan et al.*, 2008). In fact, such an approach is the only one viable in settings of such crowding or such low resolution so that detection and segmentation of individuals becomes impracticable. In some instances, a count estimate may also boost the performance of object detectors (*Rodriguez et al.*, 2011).

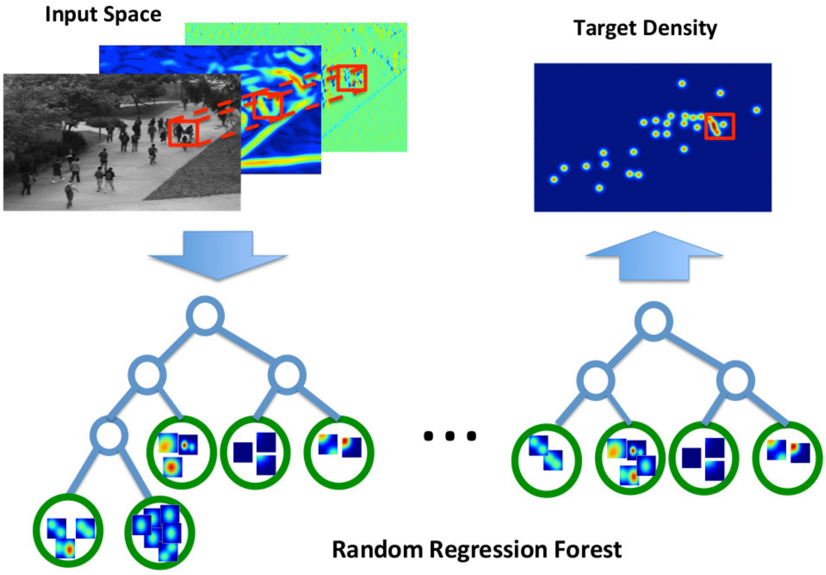


Figure 3.1: Summary of our framework. A regression random forest learns a mapping between patches in the input feature space and in the target object density space. Overlapping predictions of patches are averaged to obtain a density of objects per pixel.

An estimate of the object count N_o can either be obtained directly, by mapping from a set of global features to the real line or the integers (*Cho et al.*, 1999; *Kong et al.*, 2006; *Marana et al.*, 1997); or it can be obtained by integrating an estimated density function $F(x)$ over the image domain Ω (*density counting*)

$$N_o = \int_{\Omega} F(x)dx \tag{3.1}$$

where $F(x)$ is computed from local features. The latter approach delivers state of the art performance while requiring less training images than global regression methods. In particular, the pioneering work (*Lempitsky and Zisserman*, 2010) posits $F(x) = \mathbf{c}^T \boldsymbol{\phi}(x)$, for local features $\boldsymbol{\phi}(x)$. The weight vector $\mathbf{c}$ is learned from a training set by solving a quadratic program which minimizes the error between the true and predicted density estimates, integrated over all possible sub-windows. This model works very well, even though the predicted density can be negative in places, and the linear model requires a relatively complex set of features (BoW-SIFT *Lowe* (1999); *Csurka et al.* (2004)). A key difference between *Lempitsky and Zisserman* (2010) and our work is that *Lempitsky and Zisserman* (2010) minimize a structured loss function during training but the output of the regression is unstructured. This makes prediction at neighbouring pixels independent which can result in spatially rougher predictions as shown in Fig. 3.3. I this chapter we explore an orthogonal approach to *Lempitsky and Zisserman* (2010) because we exploit a locally structured predictor as explained in the following sections.

Our main contribution is a simplification of the original approach, arguably on the conceptual level and that of the implementation effort, and certainly in terms of computational effort in the learning phase. In particular, we compute dense features by ordinary filter banks; and propose to use a regression forest to predict entire patches of the desired density function. These structured predictions are averaged across both: predictors in the ensemble and space, akin but not identical to (*Kontschieder et al.*, 2011). In addition, we leverage the random forest out-of-bag samples to compute an uncertainty measure on the predicted object density. In section 3, we show that results compare favorably with the state of the art.

3.2 Predicting a structured regression target

Our algorithm requires a set of training images $I_i,\ i \in 1,...,N$, where all objects present in the image must be annotated with one "dot" in the center. The true density function for each pixel $x \in I_i$ is defined as a sum of Gaussian kernels centered on the user annotations:

$$F_i^0(x) = \sum_{\mu \in \mathbf{A}_i} \mathcal{N}(x; \mu, \sigma) \tag{3.2}$$

where $\mathbf{A}_i$ is the set of all annotations for image I_i and σ is a smoothness parameter. This parameter is fixed at $\sigma = 2.5$ in all reported experiments of the last section, amounting to roughly $1/4$ the objects' size. The results obtained are not very sensitive to the precise choice of σ. Since the basis functions are normalized, the overall number of objects in an image can be computed by equation 3.1.

Our key observation is that equation 3.2 represents a smooth function with a well defined local structure. As a consequence, instead of predicting the density at each location x individually, we incorporate neighborhood information by making predictions for dense overlapping patches. These overlapping predictions are then averaged in order to reduce the single pixel error in the estimate.

Rather than work on the raw images, we first compute a number ν of standard filter bank responses and then learn a nonlinear mapping

$$\mathcal{F} : \mathbf{P}_{in} \mapsto \mathbf{P}_{out} \tag{3.3}$$

from a $h \times w$ patch in the input space $\mathbf{P}_{in} \in \mathbb{R}^{h \times w \times \nu}$ to a patch in the output or target space $\mathbf{P}_{out} \in \mathbb{R}^{h' \times w'}$. Given the mapping $\mathcal{F}$, the predicted density estimate per pixel is obtained by averaging all the predicted overlapping patches:

$$\hat{F}(x) = \frac{1}{|\mathcal{P}(x)|} \sum_{\hat{\mathbf{P}}_{out} \in \mathcal{P}(x)} \hat{\mathbf{P}}_{out}(x) \tag{3.4}$$

where $\mathcal{P}(x)$ is the set of predicted patches $\hat{\mathbf{P}}_{out}$ that have pixel x in their scope. This formula works for any method that can predict not merely a single scalar, but an entire patch at a time. We have opted for regression forest (*Amit and Geman*, 1997; *Breiman*, 2001) since it offers the advantage of efficient learning and inference (around $\mathcal{O}(n \log n)$ and $\mathcal{O}(\log n)$, respectively for a fully grown tree) and high performance without parameter tweaking. In addition, it naturally leads to a confidence interval (in a loose sense) for the prediction as explained in Sect. 3.2.2.

A similar technique to Eq. (3.4) was introduced recently by *Kontschieder et al.* (2011) for image segmentation and termed *structured labels*. The authors observe that leveraging locally structured predictors obtains a similar smoothing effect to a global MRF with a reduced computational effort during inference. In that case however, discrete labels were used and neighbouring predictions where combined by taking the most predicted class in each pixel. Our approach extend that work by proposing a solution in case of real valued predictions.

3.2.1 Regression Forest using Structured Labels

Random forests have become increasingly popular in computer vision for their high flexibility, good performance and fast parallel training and testing (*Criminisi et al.*, 2011). A regression random forest is an ensemble of regression trees, each of which associates a continuous prediction with each input. Given a training set of tuples of patches $\{\mathbf{P}_{in}, \mathbf{P}_{out}\}$, we construct the trees on a randomized subset of the training examples, while the rest is kept as *out-of-bag* and can be exploited to compute a confidence interval for the output. The learning proceeds recursively, by splitting all data $\mathcal{S}_j$ arriving at a node j into a left and right subset $\mathcal{S}_L, \mathcal{S}_R$. The split is chosen by thresholding at a value τ of some simple test function f:

$$\begin{aligned} \mathcal{S}_L &= \{i \in S_j | f(\mathbf{P}_{in}) < \tau\} \\ \mathcal{S}_R &= \mathcal{S}_j \backslash \mathcal{S}_L \end{aligned}$$

At every internal node, several test functions f are randomly generated. We choose the simplest functional form of f, viz. the value of the observation at a specific pixel position and channel index inside a patch. In particular we test m_t, $m_t < \nu$, randomly selected channels at all spatial locations of the patch. Another commonly used functional form for the test function is to use the difference of two pixel at two different locations inside the patch (*Criminisi et al.*, 2011), we emulate this test function by including derivatives filter of the patch among the channels.

Given a set of test functions, the best combination of test function and threshold value is found by maximizing

$$-\sum_{i \in \mathcal{S}_R} ||\mathbf{P}^i_{out} - \bar{\mathbf{P}}^R_{out}||^2_F - \sum_{i \in \mathcal{S}_L} ||\mathbf{P}^i_{out} - \bar{\mathbf{P}}^L_{out}||^2_F \tag{3.5}$$

where $\bar{\mathbf{P}}_{out}$ is the average patch response and $||\cdot||_F$ is the Frobenius norm. Eq. (3.5) is related to the information gain of discrete classification forests (see *Criminisi et al.* (2011) for details). The recursive splitting ends when a maximum tree depth is reached (see experiments) or when $|S_j|$ is smaller than a given number. As a result of this construction, the variance of all patches associated with a node decreases with increasing tree depth, and the leaves contain clusters of similar patches. For a new test sample $\mathbf{P}_{in}$, the total response of the forest is the average of the patches stored during training time in the leaves:

$$\hat{\mathbf{P}}_{out} = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \bar{\mathbf{P}}^l_{out} \tag{3.6}$$

where $\mathcal{L}$ is the set of leaves reached by $\mathbf{P}_{in}$. As discussed in Sect. 3.3 it is possible to choose other output models in the leaves. Experimentally we found that the maximum depth of the tree is the most important parameter that influences the error of the predictor. In particular shallow trees can oversmooth the output of the regression while too deep trees tend to overfit. Fig. 3.2 gives a qualitative comparison of the prediction for various values of the maximum depth parameter for a simple 1D regression task[1].

It is to note that in contrast to ordinary scalar regression forest, in our approach, each output is now a complete patch. Fig. 3.3 shows a result of the described procedure, and a comparison with (*Lempitsky and Zisserman*, 2010)[2]. The test image in the figure was chosen to illustrate the effect of locally structured predictors. The output density of our approach and of the method of *Lempitsky and Zisserman* (2010) in Fig. 3.3 integrate to the same value 108 (true value 107), however our output appears smoother and qualitatively closer to the ground truth density.

[1] The random regression forest implementation from `http://scikit-learn.org/` is used in the 1D toy examples of this chapter.

[2] Using the implementation kindly provided by the authors, and the suggested parameters $\sigma = 4$, $N = 10$ training images, a dictionary of size 256 and $C = 0.011$.

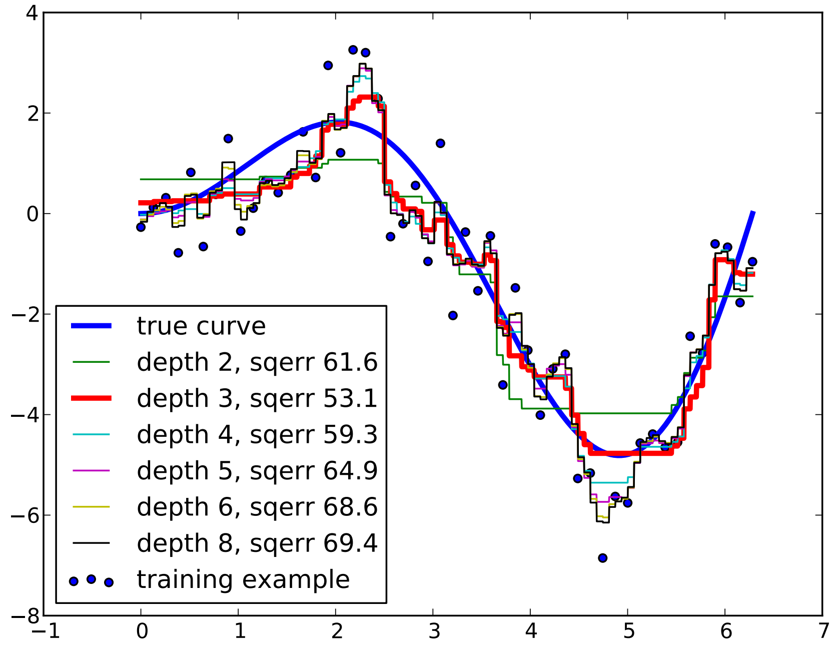


Figure 3.2: Comparison of random regression forests with identical training set and different maximum depth (all other parameters are the same) on a single dimensional regression task. An ensemble of 3 trees is used for each forest. The solid blue curve is the true target distribution. Blue dots are 50 training examples drawn from the target function and corrupted with Gaussian noise. The solid red curve is the best performing ensemble with the minimum squared error (sqerr) computed from the target function.

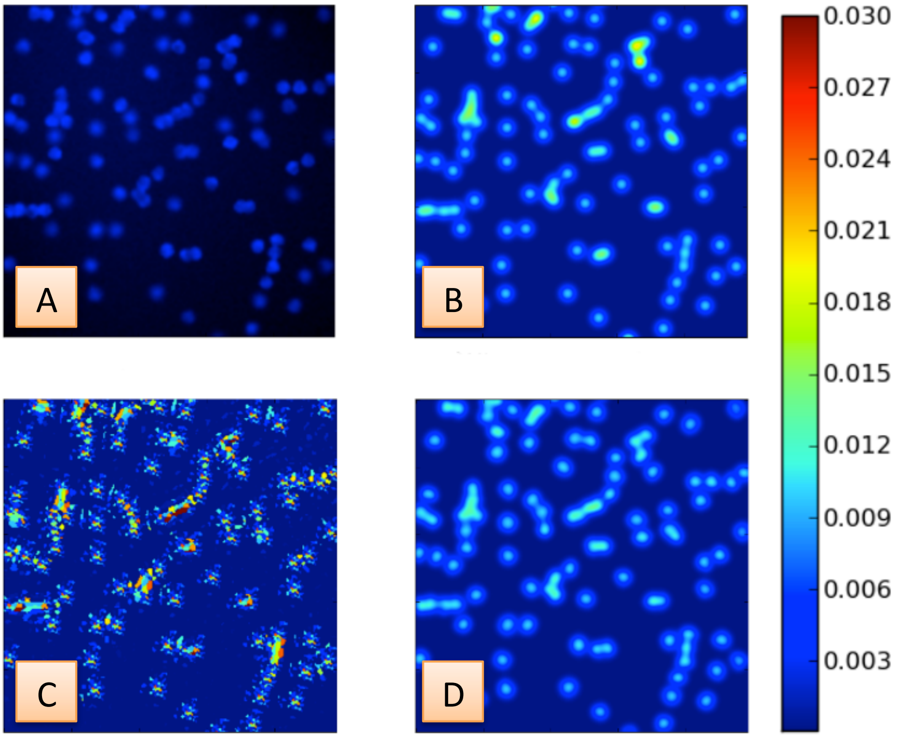


Figure 3.3: Estimated bacterial density maps. (A) row image and (B) ground truth density. (C) results obtained with algorithm (*Lempitsky and Zisserman*, 2010). (D) results obtained with our proposal. The proposed procedure makes for much smoother estimates. (Colorbar refers to all but the raw image.)

3.2.2 Uncertainty of a Prediction

Scalar quantile regression forests use all in-bag training examples in the leaves (*Meinshausen*, 2006). In our multidimensional regression case, we estimate the uncertainty of the predictions using the residual variance of both in-bag and out-of-bag samples in the leaves. Using out-of-bag samples only would seem ideal to obtain an unbiased estimate even in the small-sample case; unfortunately, not all leaves do receive out-of-bag samples. As a compromise, we push all samples from the training set (both in- and out-of-bag) down a tree and compute the total variance for each leaf l as

$$\sigma_l^2 = \frac{1}{|\mathcal{S}_l|} \sum_{i \in \mathcal{S}_l} \|\mathbf{P}_{out}^i - \bar{\mathbf{P}}_{out}^l\|_F^2 \tag{3.7}$$

For a new test sample $\mathbf{P}_{in}$, we can then assign an uncertainty measure by averaging across all trees where $\mathcal{L}$ is, again, the set of all leaves that sample $\mathbf{P}_{in}$ ends up in:

$$\hat{\sigma}^2(\mathbf{P}_{in}) = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \sigma_l^2 \tag{3.8}$$

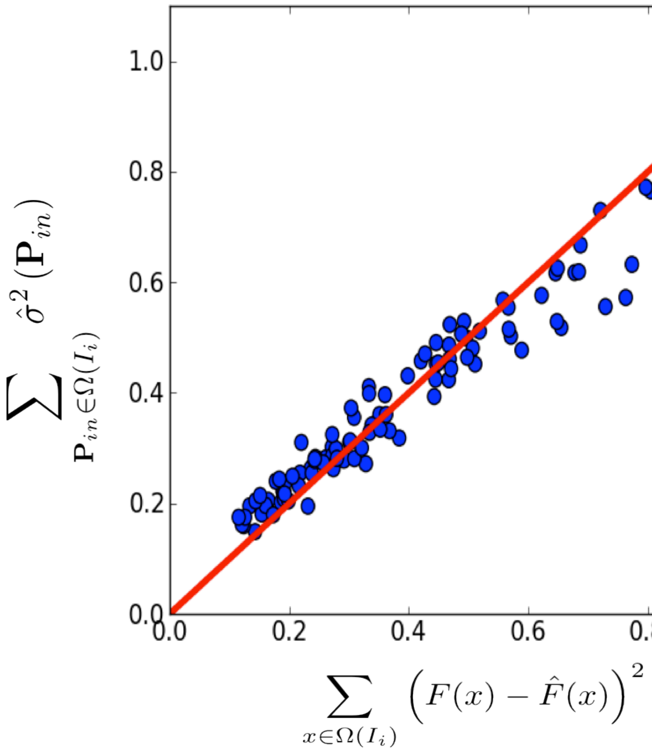


Figure 3.4: For every image of the bacterial microscopy test dataset, sum of local uncertainties vs. sum of squared local density errors. The two values show a good correlation.

Figure 3.4 shows that this uncertainty is related to the test set error, as desired.

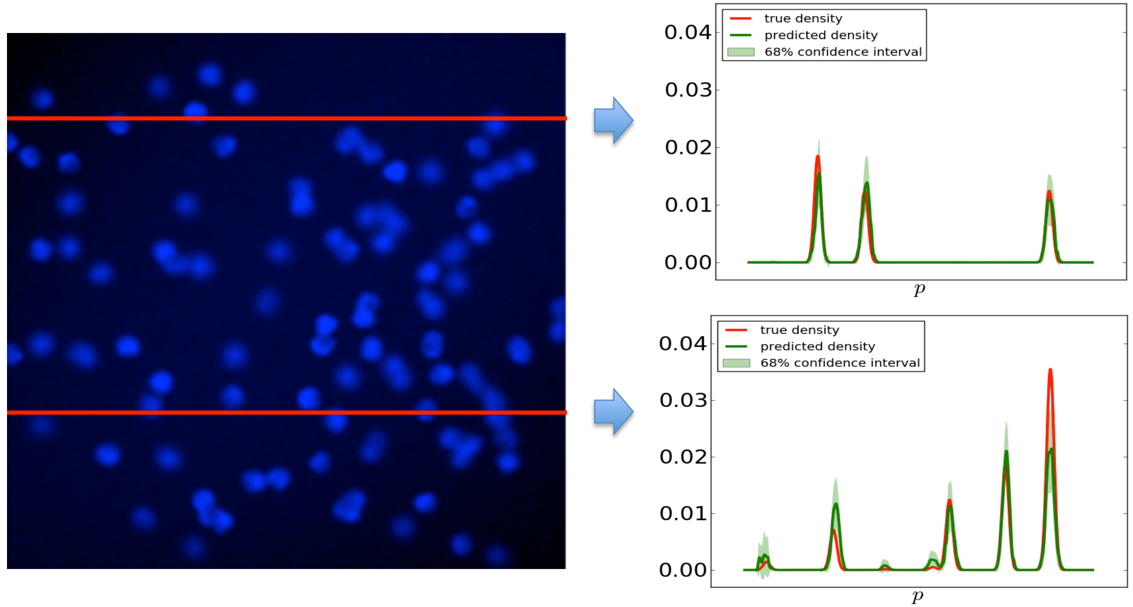


Figure 3.5: On the left raw image. On the right, two linear sections of the density corresponding to the red lines in the original image. The true density (red) is compared with the predicted density (green) and the confidence interval (pale green) is shown.

3.3 Model limitations

It is interesting to compare the output of the Random Regression forest and their uncertainty measure with other regression methods. In particular a widely used class of regression algorithms that provides a confidence bound on the prediction is Gaussian process (GP) regression(*Rasmussen*, 2006). Fig. 3.6 qualitatively compares the output of a random regression forest with the output of a Gaussian process regression for the simple task of 1D regression of the function $f(x) := x\sin(x)$. the

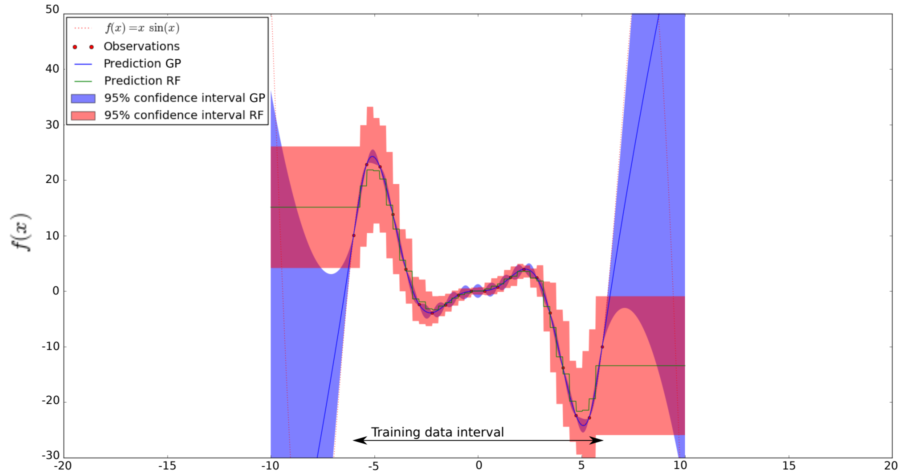


Figure 3.6: Comparison between the prediction of GP regression (solid blue line) and Random Forest (RF) regression (green line) on a 1D regression task (red line).

output of each tree is defined in Eq. (3.6) as the average of the training examples. As consequence, the prediction of the tree poorly generalizes outside of the interval of the training data. In practice all points outside of the training data interval gets assigned the same value of the closest point in the training set. More importantly, the uncertainty do not increase with the distance from the training data as it does for a GP used in the example of Fig. 3.6. These two limitations can be alleviated by choosing a more representative training set. Another possibility is fitting a probabilistic linear model in each leaf node *Criminisi et al.* (2011). This regression model in the leaf nodes often improves the performance and provides a confidence measure which decrease with the distance from training data. However, the training of the forest becomes more expensive especially in a multidimensional regression task. We found in practice that the average model is sufficient to our task provided that the training data is enough representative, we refer to *Criminisi et al.* (2011) for details on possible extensions.

3.4 Experiments

In order to compare our algorithm with the state of the art, we conduct our experiments on two publicly available benchmarks. Representative images of different object density from these two datasets are shown in Fig. 3.7. For simplicity, in the following, we use square input and output patches of same size, so $h = w = h' = w'$. An ensemble of 30 trees is used in all experiments, and the minimum split size is set to 20.

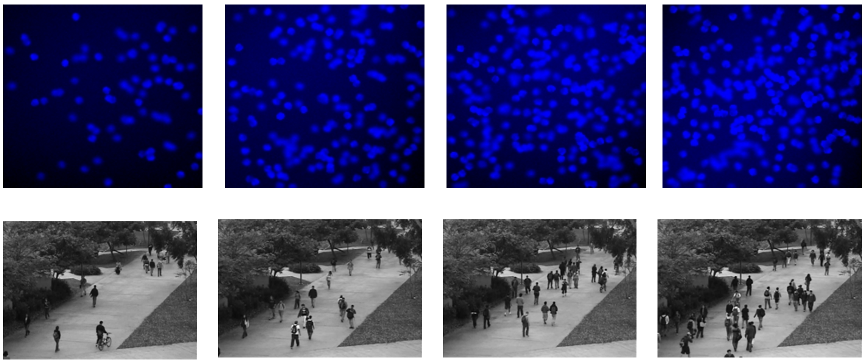


Figure 3.7: First line, representative images from dataset (*Lempitsky and Zisserman*, 2010) with increasing density of cells. Second line, representative images from dataset (*Chan et al.*, 2008) with increasing density of pedestrians.

3.4.1 Bacterial cells microscopy images

The dataset (*Lempitsky and Zisserman*, 2010) is composed of 200 simulated fluorescence microscopy images of cell cultures (171 ± 64 cells on average), with 100 images reserved for training and 100 for validation. We use a random subset of N out of all training images. For every N, we repeat the experiment five times to obtain standard errors. The maximum depth of trees is set to 10, and 3000 patches (or 1000 for N=32) are randomly sampled from each training image, with 30% of them kept out-of-bag for each tree. Following (*Lempitsky and Zisserman*, 2010), we discard the green and red channels of the raw images, and use the blue channel as well as the following features computed from it: Laplacian of Gaussian, Gaussian gradient magnitude and eigenvalues of the structure tensor at scales 0.8, 1.6, 3.2. Fig. 3.8 illustrates the input passed to the algorithm including some of the feature channels.

The results are reported in table 3.1. We also include results for pixel to pixel regression, which emerges as a special case for $h = 1$. Note that even this method uses information from a local neighborhood, through the finite width of the filters whose response is used as features for the prediction. In fact, pixel to pixel regression already performs pretty well, but extension of the regression to use and predict entire patches further improves performance and reduces variability.

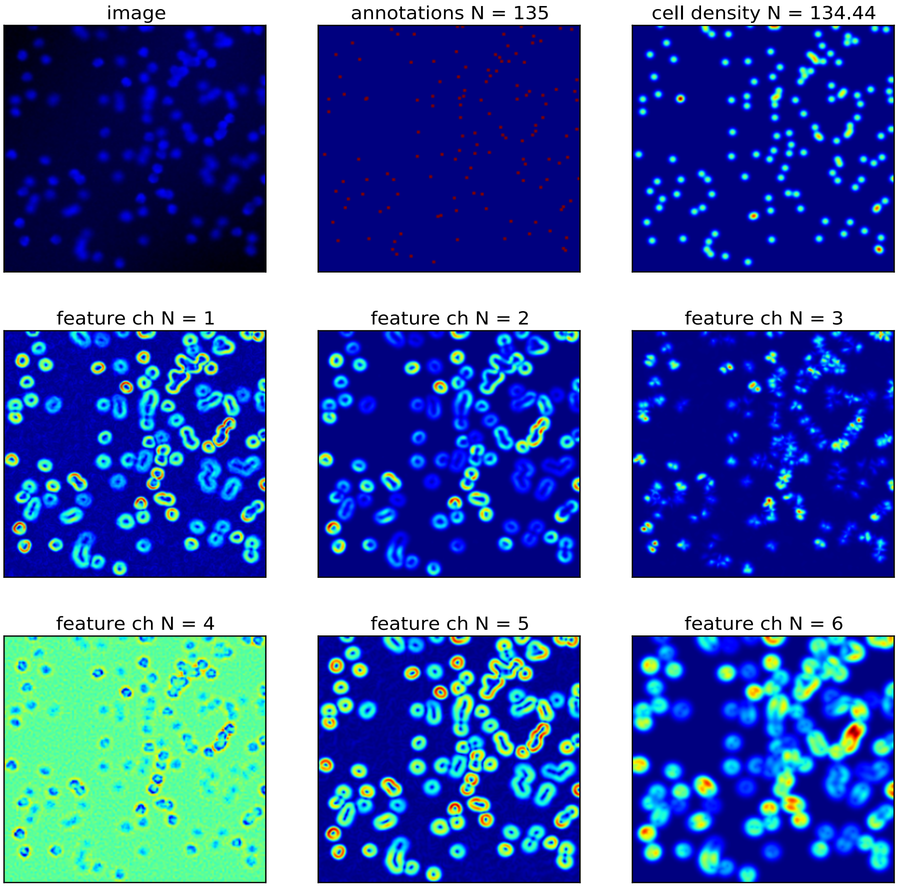


Figure 3.8: A representation of the inputs of our algorithm: original image, user annotations from which the training cell density is obtained and six of the feature channels. The integral of the training density is 134.44. This value approximates the integer count 135 due to the boundary conditions of the convolution. Comparing the original image with the feature channels, it is to note how the used features are essentially blob detectors over multiple scales. The regression forest learn a mapping between the local texture and the value of the object density in the local patch.

Table 3.1: Mean absolute errors for cell counting on microscopy images

	$N = 2$	$N = 4$	$N = 8$	$N = 32$
Detection + correction (*Lempitsky and Zisserman*, 2010)	22.6 ± 5.3	16.8 ± 6.5	6.8 ± 1.2	4.9 ± 0.5
Density MESA (*Lempitsky and Zisserman*, 2010)	5.6 ± 1.5	4.9 ± 0.6	4.9 ± 0.7	3.5 ± 0.2
This work, $h = 1$	9.1 ± 5.5	4.2 ± 0.9	3.5 ± 0.3	3.3 ± 0.3
This work, $h = 5$	7.7 ± 4.6	4.2 ± 1.1	4.4 ± 2.0	3.3 ± 0.2
This work, $h = 7$	**4.8 ± 1.5**	**3.8 ± 0.7**	**3.4 ± 0.1**	**3.2 ± 0.1**

Table 3.2: Mean absolute errors for people counting in surveillance video

	'maximal'	'downscale'	'upscale'	'minimal'
Counting-regression (*Ryan et al.*, 2009)	1.80	2.34	2.52	4.46
Counting-segmentation (*Ryan et al.*, 2009)	**1.53**	1.64	1.84	**1.31**
Density MESA (*Lempitsky and Zisserman*, 2010)	1.70	**1.28**	**1.59**	2.02
This work, $h = 7$	1.70	2.16	1.61	2.20

3.4.2 Pedestrians in surveillance video

This data set (*Chan et al.*, 2008) comprises 2000 video frames from a surveillance camera, along with dotted ground truth (29 ± 9 pedestrians on average). We compare our method with the best results obtained in the recent evaluations (*Chan et al.*, 2008; *Lempitsky and Zisserman*, 2010) following the proposed experimental protocol. Note that the "counting by segmentation" and "counting by regression" (*Chan et al.*, 2008) methods require a post-processing step to correct for large differences between the estimated counts in consecutive frames. As pre-processing, we subtract from the original images a static background as estimated by a median filter. As in the previous experiment, as features we use the raw image augmented by the filter responses described before, with the addition of a temporal derivative filter. We correct all channels for perspective distortion by simply multiplying their pixel values with the square of the provided camera perspective map. As in (*Chan et al.*, 2008; *Lempitsky and Zisserman*, 2010) we split the data into four different training and test sets: "maximal", "downscale", "upscale" and "minimal", which differ strongly in the number of training images and average number of pedestrians. We use 800 patches per image and, in order to compensate for differences in training set size, we optimized the tree depth. We note that our method performs only slightly worse than the state of the art method (*Lempitsky and Zisserman*, 2010) where a hierarchical approach, with a classifier output from the first level serving as input to a second level, was used.

3.4.3 Implementation details

The proposed approach was implemented in C++ and Cython. The multidimensional regression forest experiments are based on a modified version of `VIGRA`[3]. A demo implementation of this algorithm is provided at
`https://github.com/lfiaschi/forestcountingdemo`.

[3] `http://hci.iwr.uni-heidelberg.de/vigra/`

3.5 Discussion

In this chapter we have presented a simple and efficient method to estimate the density of objects in an image. This method offers a supervised learning strategy to object counting that is more robust when the instances are overlapping. It is based on estimating a direct mapping between the local texture and the density of objects (density-counting). This approach is appropriate for counting small, overlapping objects with similar appearance that are homogeneously distributed over an uniform background, such as cells in microscopic images. However, in our experiments, the performance decreases if the objects have high variability in size and appearance as for example in the pedestrian sequence. Another implicit assumption of this method is that the images do not contain contaminations from spurious object classes (dust, debris,other object types, *etc.*). However, when performing a cell counting assay for HCS microscopy these contaminations could be detected and filtered with the a similar method proposed in Chapter 2.

Our main contribution is to conceptually extend scalar regression random forests to output patches, a specific example of structured labels. For the task of predicting a scalar density field over the image, we obtain results on par with state of the art algorithms, while conducting all experiments with the same architecture, with simpler features and minimal parameter tuning.

Interesting directions for future work include the learning from less precise (dots are not perfectly centered), as well as from partial (image are not fully labeled) annotations. In fact, a current limitation of all counting algorithms that we are aware of (*Lempitsky and Zisserman*, 2010; *Fiaschi et al.*, 2012; *Ryan et al.*, 2009; *Chan et al.*, 2008; *Idrees et al.*, 2013; *Ma and Chan*, 2013) is the requirement of entire image labeling. Even if images could be split into smaller subregions, this training procedure is not suited to interactive refinement of the result. As the density of objects increases, for example up to a thousand cells in the image, the dotting of each instance can become a very expensive task. Moreover as the level of mutual overlap increases it is difficult even for the human expert to give a precise labelling over tight clusters of objects. To overcome these limitations, we are currently implementing a graphical user interface and specializing our algorithm to allow interactive training. Such implementation will be soon provided in the open-source framework ILASTIK[4].

The training procedure proposed in Sect. 3.2.1 minimizes the patch-wise error on the density (right hand side of Eq. (3.9)), a quantity that is only an upper bound on the counting error that measured by the difference between the target and predicted density integrated over the entire image (left hand side of Eq. (3.9)):

[4] www.ilastik.org

$$\left(\int_{\Omega} F(x)\,dx - \int_{\Omega} \hat{F}(x)\,dx\right)^2 \leq \int_{\Omega} \left(F(x) - \hat{F}(x)\right)^2 dx \tag{3.9}$$

From this observation, another interesting research direction aims at training our method from a loss function measured over the entire image. In particular, recent research have investigated the learning of Random Forest models from global loss functions (e.g.*Montillo et al.* (2011); *Schulter et al.* (2013)) using a breadth-first splitting procedure while growing the trees. It worth noticing that, in this variant of the training procedure, the left hand side of Eq. (3.9) can be evaluated for each level in the forest and may be used to guide the further growth the tree.

Finally, when analysing videos, more accurate and more localized counts (counts over small subregions rather than on the entire image) can be obtained by leveraging the count estimates at the neighboring frames. This topic constitutes the research material of the next chapter.

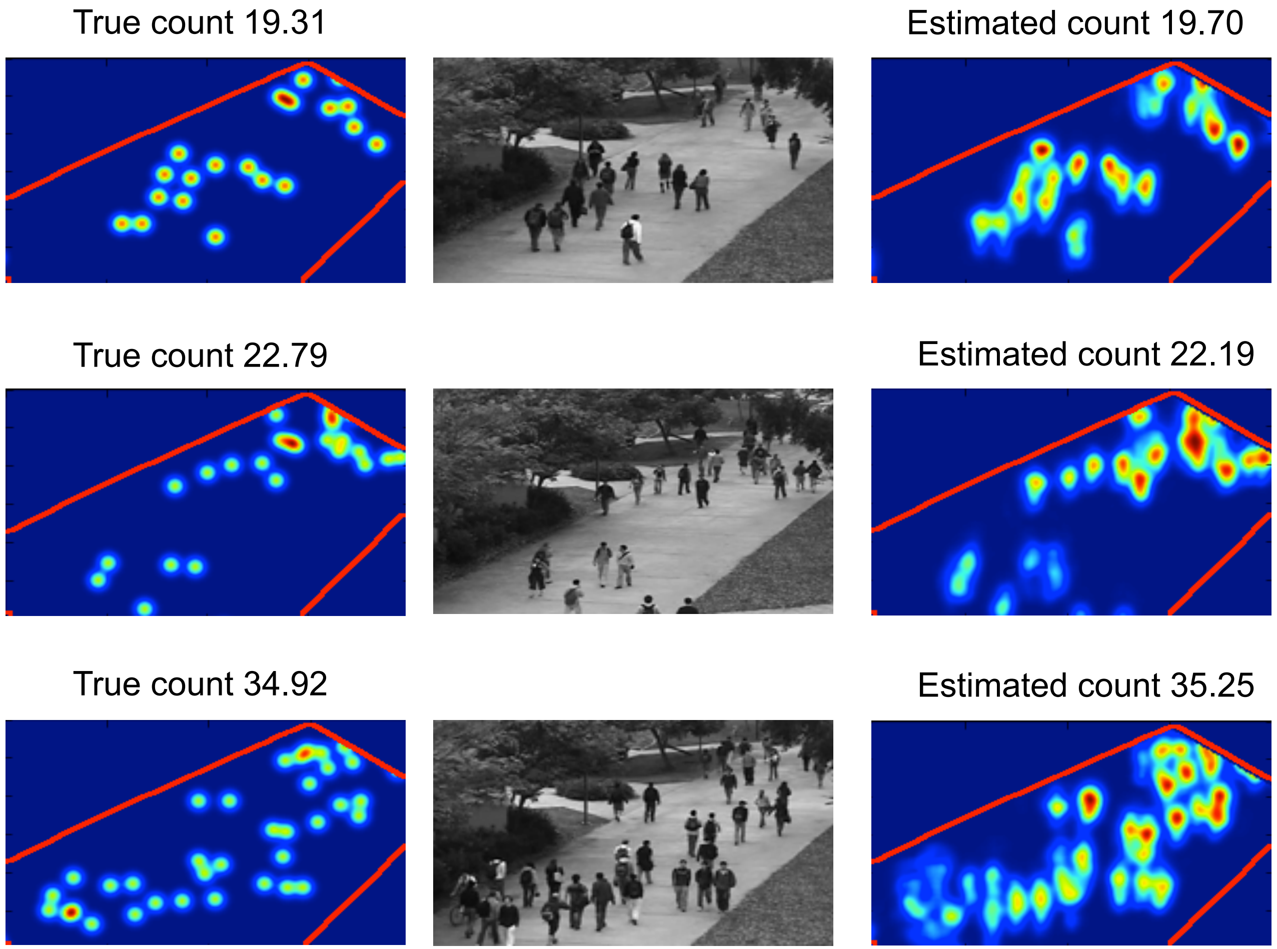


Figure 3.9: Images from the test set of the pedestrian sequence. Left column ground truth density. Central column raw images. Right column estimated density. The red line encloses the region of the road where the prediction is computed.

Chapter 4

Keeping count: leveraging temporal context to count heavily overlapping objects

Chapter 3 introduced an algorithm for counting objects in static images. That approach obtained a good average accuracy with integrating the predicted object density over the entire image, however it does not provide a good localization of the objects and moreover the accuracy strongly decreases when counts are inferred in small local regions. Relying on local information only induces a difficult regression task due to scarcity of training examples and lack of strongly predictive features. The intuition of this chapter relies on the following fact: when analyzing a sequence of images, temporal consistence in the estimates for the the object counts at neighboring frames helps ruling out ambiguous interpretations.

The main contribution of this chapter is the presentation of a novel strategy for exploiting temporal consistency when counting objects over small, spatially local regions in the frames of a movie. We develop a graphical model with deterministic higher order potentials to infer the number of objects in each foreground connected component and obtain predictions which are globally consistent across the entire video sequence. Our experiments demonstrate that global inference strongly improves over local predictions and is able to produce an accurate and coherent (i.e. consistent in neighboring frames) output with an useful runtime.

As discussed in Sect. 4.1 of this chapter, the presented work finds its motivation and application in the tracking of multiple overlapping objects (larvae of Drosophila). In fact, when tracking and segmenting multiple objects under heavy mutual occlusion, a large class of algorithms can greatly benefit from a preprocessing that reliably assesses the number of individuals in each cluster. Chapter 5 exploits the algorithm presented here as building block of a novel tracking method.

4.1 Related work and contributions

Larvae, such as *Drosophila*, are popular model organisms for behavioral studies. Significant part of the research has however focused on studying each animal individual behaviors such as feeding or locomotion (*Sokolowski*, 2001). To allow studying the social dynamics of a large population in crowded situations, the ultimate aim is to track single individuals in spite of their non-distinguishable appearance, and mutual overlap. However, this tracking scenario (represented in Fig. 4.1) constitutes a performance bottleneck of most algorithms: which may either lose one or more individuals, or confuse their identities, strongly jeopardizing the downstream analysis.

This chapter poses an important stepping stone toward the tracking of multiple Drosophila larvae by introducing a method for reliable detection of target occlusions. At the same time this technique provides an accurate estimate of the number of individuals in each connected component of the foreground (detections).

Multiple object tracking algorithms can benefit from the method presented here in two ways:

- All tracking algorithms that we are aware of require the knowledge of the number of individuals in the detections. These algorithms can be roughly divided into *tracking by detection* methods (e.g. *Bise et al.* (2011); *Kausler et al.* (2012)), which associates candidate detections at neighboring frames, and *tracking by model evolution* approaches (e.g. (*Bise et al.*, 2009; *Fontaine et al.*, 2007; *Branson and Belongie*, 2005)) which propagate templates of the tracked objects in the future frames. The first class of algorithms typically requires to set the number of tracked objects as a model parameter and often assumes that each detection contains a single object. The second class of algorithms requires initialization (seeding) of the model templates which is usually set manually or with a heuristic method (for example, detections of average size are considered as isolated individuals). When tracking hundreds of individuals manual seeding can be extremely time consuming.
- The detection of the occlusion event can be used to boost the performance of a tracking algorithm. For example, *Tang et al.* (2012) shows that it is sufficient to exploit the detection of *couples* of pedestrian walking together to improve a state of the art multiple object tracker (*Andriluka et al.*, 2010). *Henriques et al.* (2011) eliminate occluded regions from the tracking problem by matching object identities before and after the occlusion event. In addition, by explicit modeling the dynamics during of the occlusion event, it is possible to disambiguate difficult situations even when the objects appearance is not discriminative enough to distinguish the targets (c.f chapter 5).

Counting the number of objects in a small region or in a connected component of the foreground when relying only on the information present in a single image is

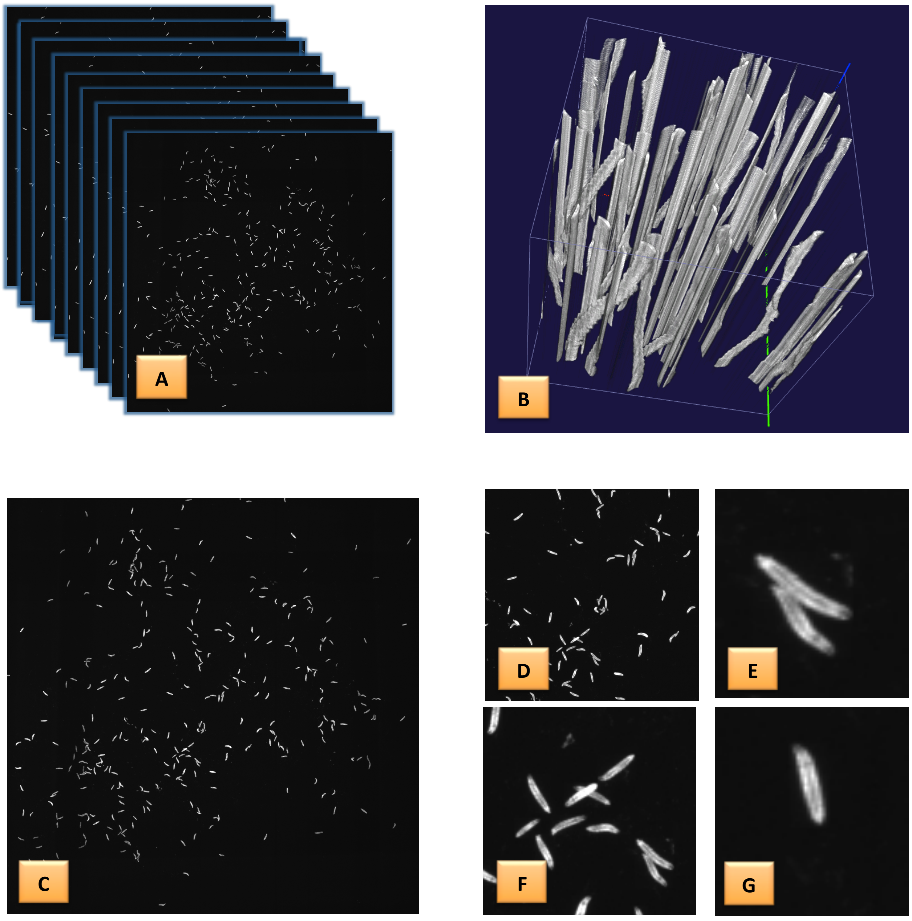


Figure 4.1: Social larvae movie dataset. (A) movie sequence frames (B) Spatiotemporal volume within a subregion of the sequence (A) (time is on the z axis). (C) One frame from the sequence (A). (D-G) Zoom into subregions from frame (C).

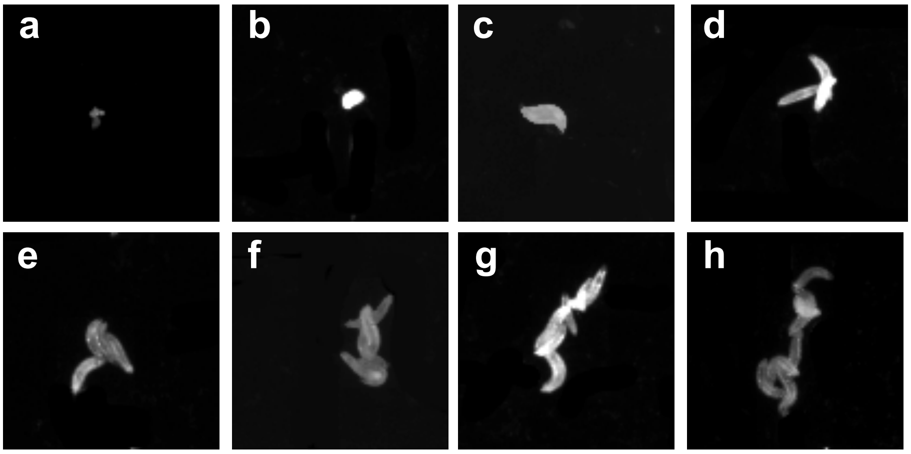


Figure 4.2: How many larvae are there in each cluster from (a)-(h)? For a human looking at the entire sequence, it is possible to confirm that the true count is: $0, 1, \dots 7$ from the top left to the bottom right. (a) False positive foreground detection. (b) Single rearing larva. (c) Two overlapping larvae. (d) Three overlapping larvae, etc.

a challenging task. First, simple features like the size of the connected component are not reliable if there is a strong overlap. Second, large clusters, which constitute the hardest cases, are less probable and thus require a time consuming mining for training examples. Third, density counting algorithms *Lempitsky and Zisserman* (2010); *Fiaschi et al.* (2012) which map the local image texture to the object density have poor performance when integrating on small image regions. Alternatively, some approaches which rely only on local information have been proposed (*Wählby et al.*, 2010; *Chan et al.*, 2008). *Wählby et al.* (2010) obtain an initial guess of the number of instances in a worm cluster from the size of the foreground region. *Chan et al.* (2008) count pedestrians learning a Gaussian process regressor starting from appearence features of the segmented region. However, in our experiments, appearance-based features that are local in space-time are not sufficient to achieve a high accuracy and therefore lead to several inconsistencies across time.

In difficult situations, as shown in Fig. 4.2, humans are still able to disambiguate and achieve a correct count by browsing the sequence forward and backward in time and looking for the separation of the individuals.

Recently, other approaches which use temporal information have been also proposed in the context of pedestrian tracking. Given the assumption that the number of visible individuals is conserved, Henriques et al. (*Henriques et al.*, 2011) obtained the count of pedestrians in merged detections as a minimum cost flow over the detection graph. However, their method requires manual initialization of isolated individuals, and relies on the size of the foreground detection only, which is a weak local cue in case of overlapping objects.

Our approach is to mimic the strategy of the human expert by using a graphical model with deterministic constraints. Our main contribution is to integrate multiple local cues to achieve an object count that is consistent across space and time. The coupling of all estimates across space and time allows to propagate information from simpler parts of a video to complex clusters of larvae.

The relation between constraint satisfaction problems and graphical models has been thoroughly investigated in the context of Bayesian networks (*Mateescu and Dechter*, 2008) where several dedicated inference techniques have been proposed. Akin (*Bise et al.*, 2011; *Kausler et al.*, 2012; *Roth and Yih*, 2007), we formulate the MAP inference step as an integer linear program (ILP) that is solved with a standard software package CPLEX[1]. We show that this formulation can handle a large volume of high resolution data. Our experimental evaluation demonstrates accurate and coherent results which can be exploited by downstream tracking and segmentation algorithms.

4.2 Methods

4.2.1 Model definition

Our workflow is depicted in Fig. 4.3. Firstly, we produce a set of candidate segmented foreground regions (N in total) as detailed in Sect. 4.3.1. Secondly, for each region i, we assign a random variable $x_i \in \{0, ..., M\}$ expressing the number of contained larvae. We include the label $k = 0$ since we want to handle cases in which debris and other contaminations are segmented as foreground. M represents the maximum number of objects in a foreground region. Thirdly, as detailed in Sect. 4.3.1, we link neighboring foreground regions from consecutive timesteps in order to build an undirected detection graph $G = (\mathbf{X}, \mathbf{E})$, where $\mathbf{X} = \{x_i\}_{i=1}^{N}$ is the collection of all random variables and $\mathbf{E} = \{e_i\}_{i=1}^{L}$ is the set of edges. Our assumption that larvae cannot enter or leave the field of view[2] naturally imposes a set of constraints that relates all variables in $\mathbf{X}$. To make the structure of the problem explicit, we define the factor graph $G' = (\{\mathbf{X}, \phi, \Phi\}, \mathbf{E}')$ where two types of potentials are present (as depicted by small and big black squares in the central part of Fig. 4.3). First order potentials $\phi(x_i, f_i)$ express our belief that the current region contains a certain number of larvae based only on a local feature vector $\mathbf{F} = \{f_i\}_{i=1}^{N}$; higher order potentials are deterministic functions of the variables in their scope that enforce consistency. We define $G_{t,t+1} \subseteq G$ as the subgraph comprising the variables at time t and $t + 1$, and $S^j_{t,t+1}$ as the j^{th} connected component of this subgraph:

$$G_{t,t+1} = \bigcup_j S^j_{t,t+1} \quad ; \quad S^j_{t,t+1} \cap S^k_{t,t+1} = \emptyset \quad \forall j \neq k$$

[1] `http://www-01.ibm.com`

[2] Except for the borders of the image. Boundary conditions are explained in Sect. 4.3.1.

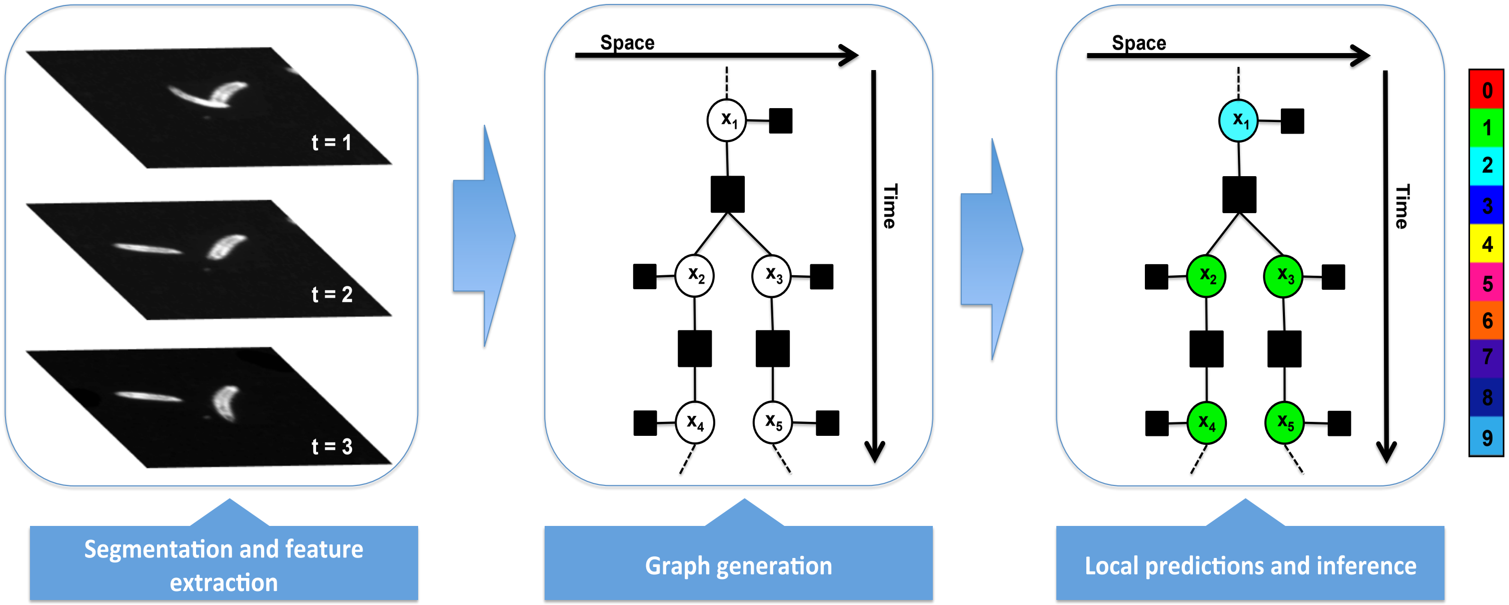


Figure 4.3: An overview of the workflow. Each image is first thresholded into foreground vs. background. Then, a random variable representing the object count is associated with each spatially connected component. The probability distribution over the true object count is estimated, based on purely local features extracted from each connected component, by unary factors (represented by small black boxes). The random variables are also connected in time by deterministic higher order factors (large black boxes) that encapsulate the "conservation of objects" constraints. The connectivity of these higher order factors follows from spatio-temporal overlap. Finally, global inference determines the consensus estimate of the true object count.

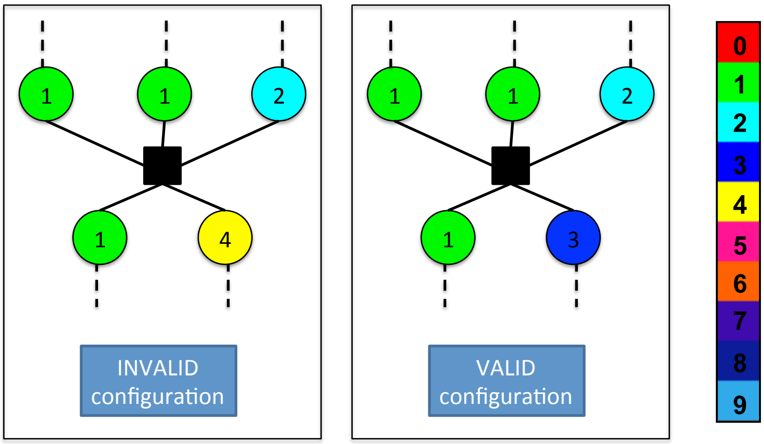


Figure 4.4: Valid versus invalid configuration of a subgraph of 5 variables. Each variable x_i is represented by a circle, the colors code the values assigned to the variables according to the palette on the right. Black squares are the higher order potentials relating the variables at neighboring timesteps. On the left an invalid configuration (the sum of the three upper variable values is bigger then sum of the two lower variable values), on the right a valid configuration.

The higher order factors are defined as follows:

$$\Phi(\mathbf{X}_t^j, \mathbf{X}_{t+1}^j) = \begin{cases} 0 & \text{for } \sum_{x_i \in \mathbf{X}_t^j} x_i - \sum_{x_i \in \mathbf{X}_{t+1}^j} x_i = 0 \\ \infty & \text{for } \sum_{x_i \in \mathbf{X}_t^j} x_i - \sum_{x_i \in \mathbf{X}_{t+1}^j} x_i \neq 0 \end{cases} \tag{4.1}$$

where $\mathbf{X}_t^j \in S_{t,t+1}^j$ is the set of variables at time t in the j^{th} connected component. The total energy can be expressed in terms of the factor graph as:

$$U(\mathbf{X}) = \sum_{i=1}^{N} \phi(x_i, f_i) + \sum_t \sum_{S_{t,t+1}^j \in G_{t,t+1}} \Phi(\mathbf{X}_t^j, \mathbf{X}_{t+1}^j) \tag{4.2}$$

The Gibbs relation $P(\mathbf{X}, \mathbf{F}) = \frac{1}{Z} e^{-U}$, where Z is the partition function, allows to establish the equivalence between the MAP configuration of the associated probabilistic graphical model and the argmin of Eq. (4.2). Therefore higher order potentials assign 0 probability mass to forbidden variables configurations. A visual example of a forbidden configuration is given in Fig. 4.4.

Unlike previous tracking approaches (e.g. *Bise et al.* (2011); *Kausler et al.* (2012)) our proposal contains unary potentials and only deterministic higher order potentials, without any smoothness term relating neighboring variables. In such a model inference can be difficult for most of available solvers, however we rely on a generic technique described in the next section.

4.2.2 Integer linear program formulation

We implement the MAP inference as an integer linear program with indicator variables. A recent experimental study on inference techniques (*Kappes et al.*, 2013) have demonstrated that this formulation is able to provide global optimal solution by exploiting advanced branch and bound techniques. The size of the inference problems which have been solved with this method is of the order of $10^5 - 10^6$ variables (e.g. *Andres et al.* (2012)).

To tansform problem Eq. (4.2) into an integer linear program (ILP), to each foreground region $i = 1, \ldots, N$ are associated the binary indicator variables $x_i^k \in \{0,1\}$, $\sum_{k=0}^{M} x_i^k = 1$ encoding the number of objects in the region. We set the unary potential to be:

$$\phi_i^k := -\log p(x_i^k = 1|f_i) \tag{4.3}$$

where $p(x_i^k = 1|f_i)$ is the probability for component i to contain k object instances given the features, as estimated by a local classifier (cf. Sect. 4.2.3). Then we have:

$$\begin{aligned} &\min_{x_k^i} \sum_{k,i} \phi_i^k x_i^k \quad \text{s.t.} \\ &x_i^k \in \{0,1\} \quad \forall i,k \\ &\sum_{k=0}^{M} x_i^k = 1 \quad \forall i \\ &\sum_{k} \sum_{x_i^k \in \mathbf{X}_t^j} k x_i^k - \sum_{k} \sum_{x_i^k \in \mathbf{X}_{t+1}^j} k x_i^k = 0 \ \forall S_{t,t+1}^j \in G_{t,t+1} \ \forall t \end{aligned} \tag{4.4}$$

The first two constraints ensure that the possible states of the random variable x_i are mapped to a set of binary indicator variables $\{x_i^k\}$. The third constraint represents the conservation of the number of larvae as enforced by the higher order potentials of Eq. (4.1).

4.2.3 Unary potentials

For reasons both of accuracy and tractability, it is important to use informative unary potentials. The probability $p(x_i^k = 1|f_i)$ is learned with a classifier from labelled training data. However, this is a strongly unbalanced classification problem since, at least in our data, there are much fewer training examples available for classes $k > 1$. Therefore, we choose the following strategy: firstly, we train a classifier with examples of label $0, 1, 2, 3, many$, where class labelled "$many$" includes all examples of foreground regions containing 4 or more individual instances. Secondly, we take the probability $p(x_i^{many} = 1|f_i) := \beta_i$ and distribute it among the classes $k \geq 4$ according to the parametric function:

$$p(x_i^k = 1|f_i) = \beta_i \exp\Big(- \frac{(\tilde{\tau} - k)^2}{2\sigma^2} \Big)/\alpha \ , \ k \geq 4 \tag{4.5}$$

Here $\tilde{\tau}$ is the size of the foreground region (in pixels) normalized by the average size of all observed foreground regions and α is a normalization constant. In the following experiments we fix $\sigma = 2$. This procedure allows us to obtain a very sharp unary term for foreground regions containing few larvae, while we rely mostly on the global inference step to find the number of instances in big clusters.

4.3 Experimental results

4.3.1 Data and preprocessing

A population of 72 hours old *Drosophila* larvae was filmed for 5 minutes with a temporal resolution of 3.3 frames per second, 1000 frames in total. Images have a spatial resolution of 135.3 μm/pixel, a size of 1560×1600 pixels, and contain on average 323 larvae[3].

For detection and segmentation of the foreground regions we use the open-source software ILASTIK (*Sommer et al.*, 2011). As explained in the following it is important to obtain an accurate segmentation, therefore we train the segmentation algorithm with 20 images from the original sequence (every 50 timestep). It is important to note that ILASTIK allows interactive refinement of the segmentation, therefore only few label strokes over the images selected for training are needed in practice. After elimination of tiny isolated objects ($\tau < 15$ pixels), we compute connected components of each thresholded foreground probability image of the series (threshold set to 0.5 probability of the foreground). Given the high temporal resolution of this dataset, which is demonstrated in Fig. 4.1B, the graph G is created by linking foreground regions from neighboring timesteps that overlap spatially, by more than 10 pixels. A nearest neighbor approach could be used for lower temporal resolution data. To avoid errors at the boundary of the image we exploit a simple heuristic while constructing the graph. All foreground regions which are not fully inside a safety margin of 100 pixels from the image borders are excluded to avoid dealing with truncated larvae (cluster).

4.3.2 Training unary potentials

As explained in section 4.2.3, we obtain the unary term partially from labelled training data and partially with the parametric function 4.5. All images in the sequence were labelled by hand by an human expert to establish a gold standard. An interactive labeling tool was developed for this task to give the human expert the possibility to scroll back and forth the sequence before assigning a particular label.

For the training of the classifier (a standard random forest (*Breiman*, 2001)) we use only 10 images from the first 250 timesteps of the sequence (every 25 timesteps, 1% of the available data). A set of 21 object features describe the size and the shape

[3] Larvae can exit from the field of view of the microscope. Again, boundary conditions are explained in Sect. 4.3.3

of each foreground region: area, convex area, eccentricity, equivalent diameter, axis lengths, perimeter, solidity, mean intensity, variance of the intensity, total intensity and the magnitude of the first 10 Fourier contour descriptors (cf. (*Jähne*, 2002) for an extensive review). The analysis of feature importance from the random forest revealed a dominant weight of the Fourier descriptor compared to the other features. The intuition about this fact is that a is a useful cue to infer the number of larvae is related to the frequency of the kinks in the contour of cluster. In our implementation the Fourier descriptors were not normalized.

4.3.3 Implementation details

In our experiments, we run and evaluate the results on the entire sequence (1000 timesteps). Inference takes 28 minutes. We use the following optimizations: firstly, not every larva interacts with all others over the course of time, therefore we can solve the optimization problem separately for independent connected components of the factor graph. We found our dataset to contain 158 subgraphs, with one huge subgraph consisting of 266215 foreground regions accounting for 93% of all regions (solving the problem for this subgraph takes approximately 14 min). Secondly, we use CPLEX's warm-start interface to initialize the solver with the assignments obtained by minimizing the unaries. For most of the subgraphs (not the biggest one) this initialization delivers already the optimal problem solution. Thirdly, we add conservative constraints that rule out improbable assignments. This heuristic greatly reduces the search space of the branch and bound algorithm and is fundamental to obtain the solution in such a short runtime. In particular, if τ is the size in pixels of the foreground region, for $\tau < 50$ we add the constraint $x_i \leq 1$. In a similar vein, $\tau < 120 \Rightarrow x_i \leq 2$, $\tau > 50 \Rightarrow x_i \geq 1$, $\tau > 300 \Rightarrow x_i \geq 2$. The value for these constraints was chosen after manual inspection of few cluster examples.

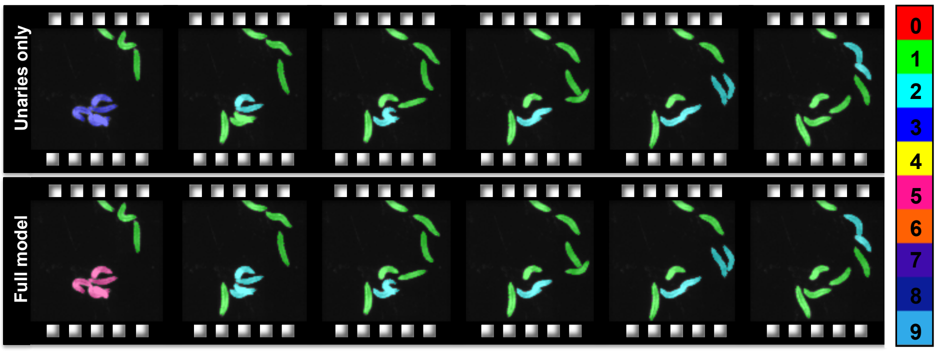


Figure 4.5: Comparison between the predicted counts obtained by simple minimization of the unary potentials (first row) and the proposed MAP solution (second row). The second row shows the consistent, and correct, labelling (in particular first and second frame).

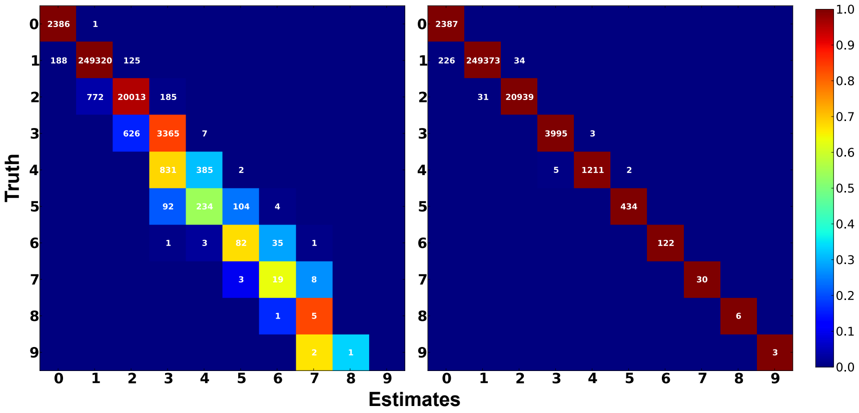


Figure 4.6: Comparison between the confusion matrix from simple minimization of the unary potentials (left) and the proposed MAP solution (right). The false colors reflect the confusion after row-wise normalization.

4.3.4 Results

Figs. 4.5 and 4.6 demonstrate the consistent improvements of the full model over local predictions. In particular, Fig. 4.5 illustrates how the higher order potentials help disambiguating an inconsistent assignment. The evidence from the right part of the sequence is propagated to the initial timesteps in order to correct the local errors. Fig. 4.6 shows that the strongest improvements are obtained for cluster of ≥ 3 larvae. Indeed the parametric function 4.5, which only depend on the size, underestimates the count for larger clusters. While a better choice of the parametric function could be in principle taken, this is not necessary as shown on the right hand side of Fig. 4.6. The full model is almost perfectly able to correct for the errors in big clusters of larvae.

To summarize our findings with a single number, we define an adjusted precision score for regions containing three or more individuals:

$$rec = \frac{\bar{\mathrm{TP}}}{\mathrm{TP} + \mathrm{FP}} \tag{4.6}$$

where TP and FP are the true positives and false positives. $\bar{\mathrm{TP}}$ is similar to TP but only considers foreground regions that do not directly violate consistency constraints as in Eq. (4.1). For the full model rec is equivalent to standard precision, while for the reduced model rec penalizes temporally inconsistent assignments. The rational behind this measure is that a temporally inconsistent assignments could be dangerous to a downstream tracking algorithm by creating inconsistent situations that might be difficult to model explicitly. Under measure Eq. (4.6), the score of the prediction obtained by simply minimizing the unaries is 61.1%, while the score of the full model is 99.8%. While this result is close to be perfect, sources of errors remain due to under-segmentation before the construction of the graph as discussed in the next section.

The spatiotemporal plot of the obtained trajectories is shown in figure Fig. 4.8. Green trajectories correspond to isolated individuals. This plot demonstrates that most of the larvae are isolated for a long time, akin the ideal gas model in physics, we termed this time interval the larva mean free path. A perfect tracking is obtained for the free larva path between occlusion events. Ambiguous regions are limited in space and time to subgraphs corresponding to a group of larvae occluding each other (Fig. 4.8 right). A novel method to resolve these occlusion events is presented in chapter 5.

4.3.5 Remaining error sources

The higher order potentials enforce temporal consistency with respect to the spatiotemporal graph, however some deviations from the gold standard are still possible even in a consistent prediction. From Fig. 4.6 we see that the most confused classes are labels 1,2 and labels 3,4. Fig. 4.7 illustrates two prototypical situations where the higher order potential are insufficient to infer the correct assignments for the random variables. In Fig. 4.7(A) because of an under-segmentation error a foreground region corresponding to one larva (pale orange circle) is removed from the graph. The missing link causes the deterministic potentials to produce an error that propagates to the variables in the following timesteps (orange circles). In Fig. 4.7(B) the unary potentials are ambiguous, however the higher order potential is respected. In this situation, it is not possible to disambiguate the case where the two connected components in orange flip their label. Problem (A) could be fixed by introducing into the model the possibility for a larva to disappear. A smoothness cost between clusters assignments at neighboring timesteps could alleviate problem (B).

It is important to note that our work builds on two assumptions which allow the introduction of the deterministic constraints: first, that the detection is sufficiently sensitive such that an isolated object cannot disappear for short periods of time. Second, that the temporal resolution of the data is sufficient for the construction of the consistency graph. Violations of the first assumption can introduce temporally confined errors as in our experiments, or could produce more relevant mistakes that are propagated over time by the deterministic constraints.

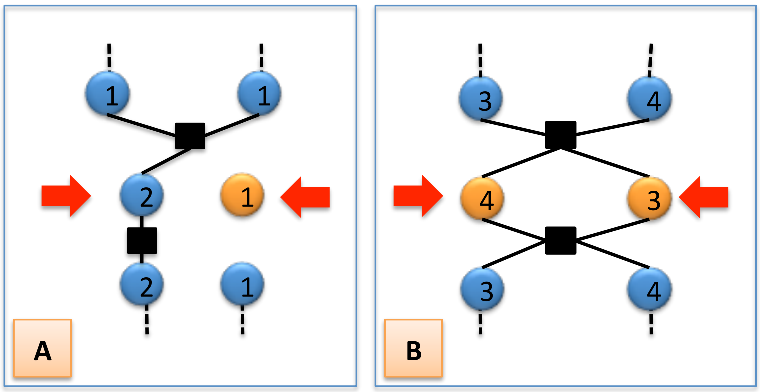


Figure 4.7: An illustration of two errors which are temporally consistent and cannot be avoided with the proposed deterministic potentials. Each circle is a random variable. The number inside each circle indicates the state of the variable. Higher order potentials and edges of the detection graph are indicated in black. Red arrows point out variables which are wrongly assigned compared to the gold standard.

4.4 Discussion

This chapter poses a stepping stone towards the tracking of a large population of Drosophila larvae. In particular, we have proposed a method for reliable detection of mutual occlusion that also provides an estimate of the number of individuals in the clusters of larvae.

Unlike static object counting approaches, discussed in chapter 3, that use only information with a limited spatio-temporal support, we introduce a graphical model to couple predictions at neighbouring frames. For the counting task, the main contribution of this chapter is to demonstrate that the introduction of higher order deterministic consistency constraints outperforms local predictions.

This algorithm could already support behavioral biologists by producing two measurements. First, we allow estimating the rate of larvae encounters and second, we provide the tracking of the objects between the mutual occlusion events. Yet this tracking scheme cannot recover the identity of single individuals which are lost during occlusion. The principal benefit, however, is to provide input for future algorithms that should allow the tracking of each and every larva, even through complex agglomerates. In fact, the method presented in this chapter allows decomposing the large tracking problem for the entire sequence and the entire population into ambiguous and unambiguous spatiotemporal regions comprising only few animals. Ambiguous regions correspond to mutual visual occlusion of the targets. We are able to resolve these regions and disambiguate the single animal identities by the approach described in the next chapter.

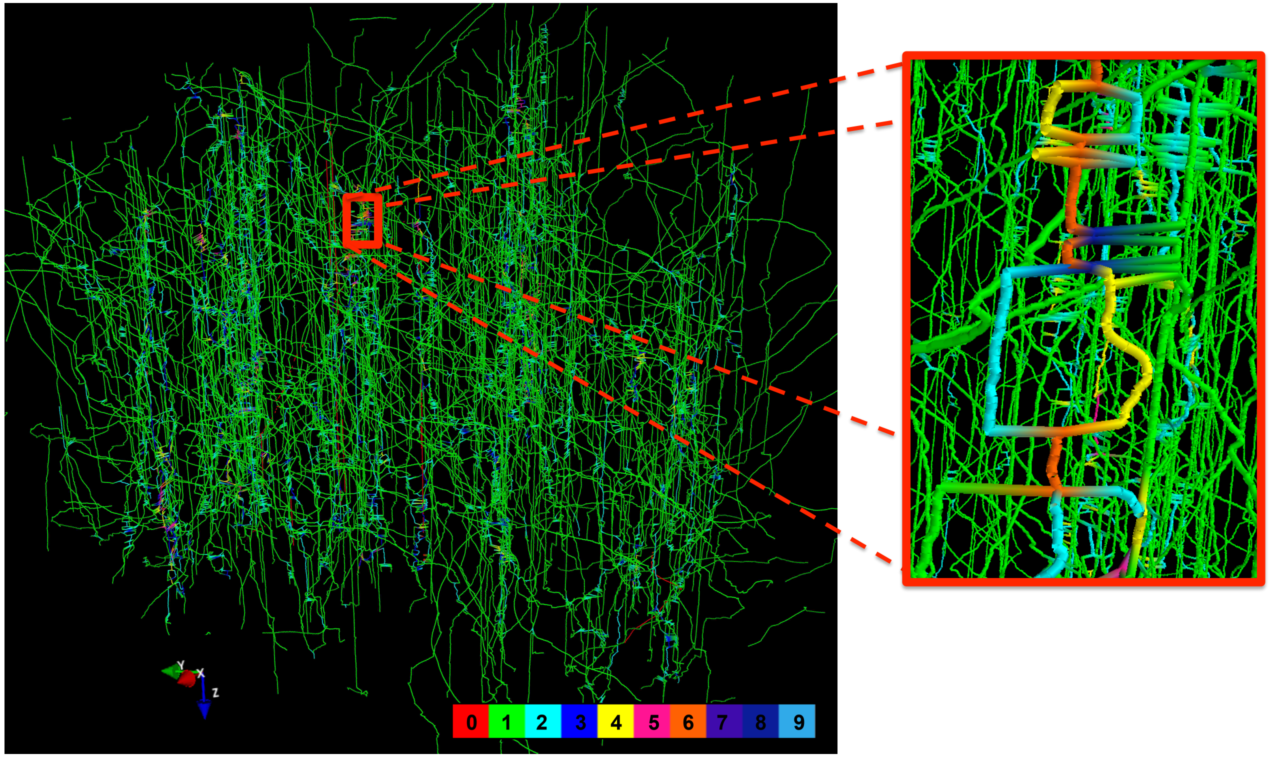


Figure 4.8: Larvae clusters and count in spacetime (time is along the z axis). Colors are according to colormap on the bottom right. We can interpret the trajectories as water pipes, color indicates different capacities and the flow is conserved at the junctions. Green represents isolated larvae whose tracks are an intermediate output of the proposed algorithm. Most of the larvae remain isolated for long period of time (green lines) their trajectory constitute an intermediate result of our algorithm.

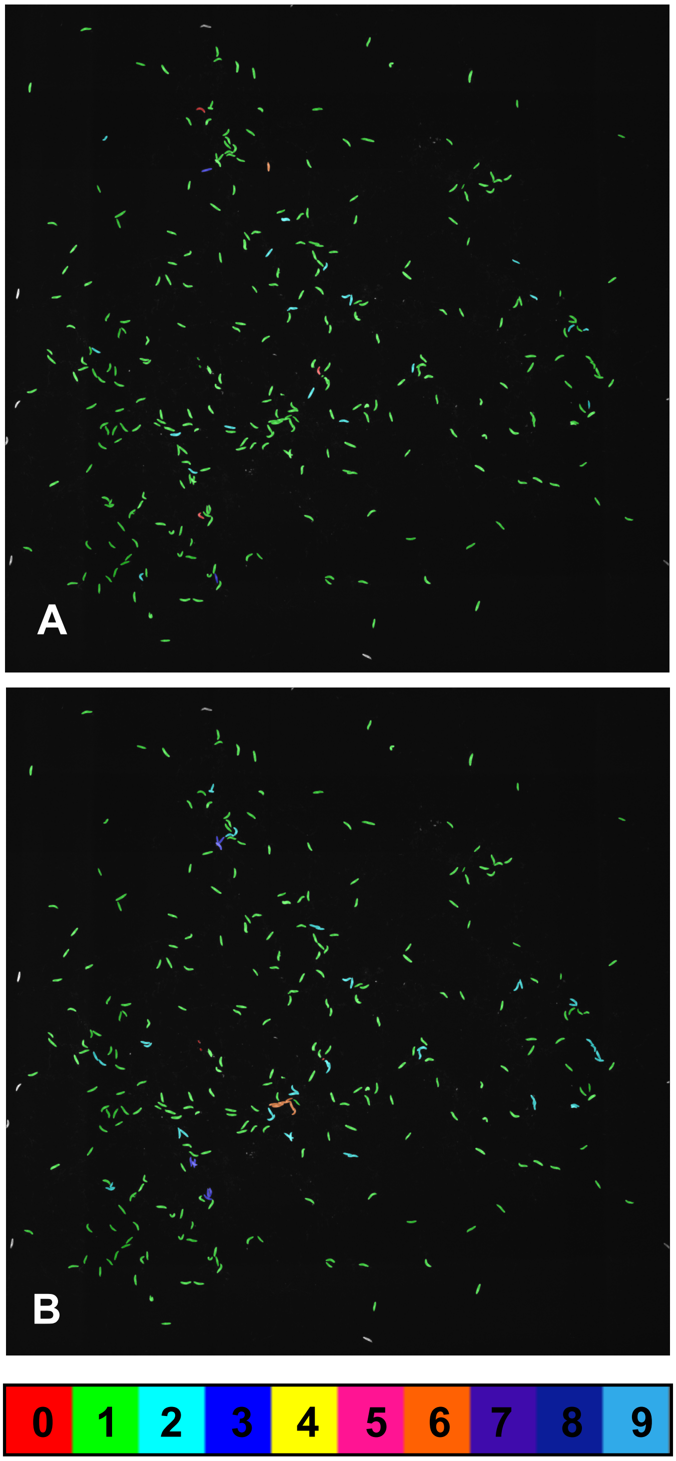


Figure 4.9: Qualitative comparison between the prediction from the unaries (A) and the full model (B) on a test frame of the sequence. The inferred larvae count is color coded according the palette given at the bottom of the figure. In particular, several big isolated larvae (in cyan in figure (A)) are misclassified by the unaries as clusters of two larvae. A tight cluster of 6 larvae (in orange in figure (B)) is mistakenly classified by the unaries as a single larva.

Chapter 5

Learning to disambiguate indistinguishable objects over time: structured learning from partial annotations

The research presented into this chapter aims at providing a method to elucidate the social dynamics of larvae of *Drosophila*. To this end, the tracking of multiple, almost indistinguishable, translucent and deformable objects is required. In addition, these objects can undergo heavy mutual occlusion and overlap for many frames. Such a challenging task is approached in two steps. Chapter 4 has proposed a novel method to detect the mutual occlusion events. This chapter builds upon that work by restricting the focus to disambiguate the spatiotemporal regions where occlusion takes place. This two steps approach reduces the tracking problem for the entire population to smaller sub-problems comprising only few individuals.

Mutual occlusion is a performance bottleneck for most tracking algorithms. The task of resolving the object identities *after* occlusion is often left to a classifier relying on specific appearance features. In this chapter we take a different approach. The key idea presented here is to model the dynamics *during* the occlusion event. We propose such a model for the challenging situation of multiple indistinghuishable but translucent objects. Thus we introduce a novel convex formulation which handles complex motions and non-rigid deformations, by jointly optimizing the flows of multiple latent intensities across frames. The model hyperparameters are learnt from training data in such a way to provide an accurate description of the intensity flows of the translucent objects while they are crossing each other. However, the flows are not directly observed variables (i.e. latent variables), hence the user cannot provide their labels. For this reason, we propose a learning method based on weakly supervised structured learning that exploits *partial* user annotations only. The strength of our framework is to disambiguate the identity of the objects during and after occlusions without assuming that the targets have different appearance. In turn this allows reconstructing the tracking for each individual object for the entire movie.

The rest of this chapter is organized as follows: first, we show the relations between our work and the previous literature and define our contributions. Second, we introduce our inference and learning procedure and motivate the key ideas of our approach. Finally, we present and discuss the results of extensive experiments. The approach is validated on a challenging dataset for multiple Drosophila larvae tracking. Our method tracks multiple larvae in spite of their poor distinguishability and minimizes the number of identity switches during prolonged mutual occlusion.

5.1 Introduction and contributions

In recent years, numerous brilliant studies are born from the happy marriage between behavioral biology and sophisticated multiple-object tracking algorithms, e.g. *Branson et al.* (2009); *Branson and Belongie* (2005); *Khan et al.* (2006). Our research focuses on the social dynamics of larvae of *Drosophila* which is a popular model organism for behavioral studies. These experiments require tracking of multiple individuals in spite of their non-distinguishable appearance and mutual overlap. In particular, two aspects violate the assumptions of state of the art tracking algorithms which exploit object appearance features *Nillius et al.* (2006); *Henriques et al.* (2011); *Zhang et al.* (2008); *Shitrit et al.* (2013); *Jiang et al.* (2007) and/or motion models *Okuma et al.* (2004); *Collins* (2012) in order to handle occlusion. Under our experimental condition:

- Larvae are poorly distinguishable, translucent and mostly untextured objects of deformable shape.
- Larvae, when in contact, can exhibit an erratic motion.

Our formulation handles complex motion and non-rigid deformations without assuming the tracked objects to be dissimilar. At the core of our approach, we model the observed intensities as a mixture of the latent intensities of each individual object. Thus, we disambiguate object identities by jointly optimizing the movement of multiple latent flows of intensity masses. This leads to a highly flexible model with many parameters that balance costs derived by multiple generic low level cues. An important difference between our approach and *Nillius et al.* (2006); *Henriques et al.* (2011); *Zhang et al.* (2008); *Shitrit et al.* (2013); *Jiang et al.* (2007); *Okuma et al.* (2004); *Collins* (2012) is that we exploit structured learning from partial annotations *Lou and Hamprecht* (2012) to learn the hyperparameters, rather than rely on manual tuning. Our main contributions are:

- A novel linear formulation of the multiple objects tracking problem. Our model is targeted to model the occlusion event of overlapping identical translucent objects.
- A supervised learning strategy to parameterize the corresponding energy terms based on partial annotations, which requires a minimal user effort during labeling.
- A solution to low density tracking of *Drosophila* larvae which minimizes the number of identity switches during prolonged occlusion.

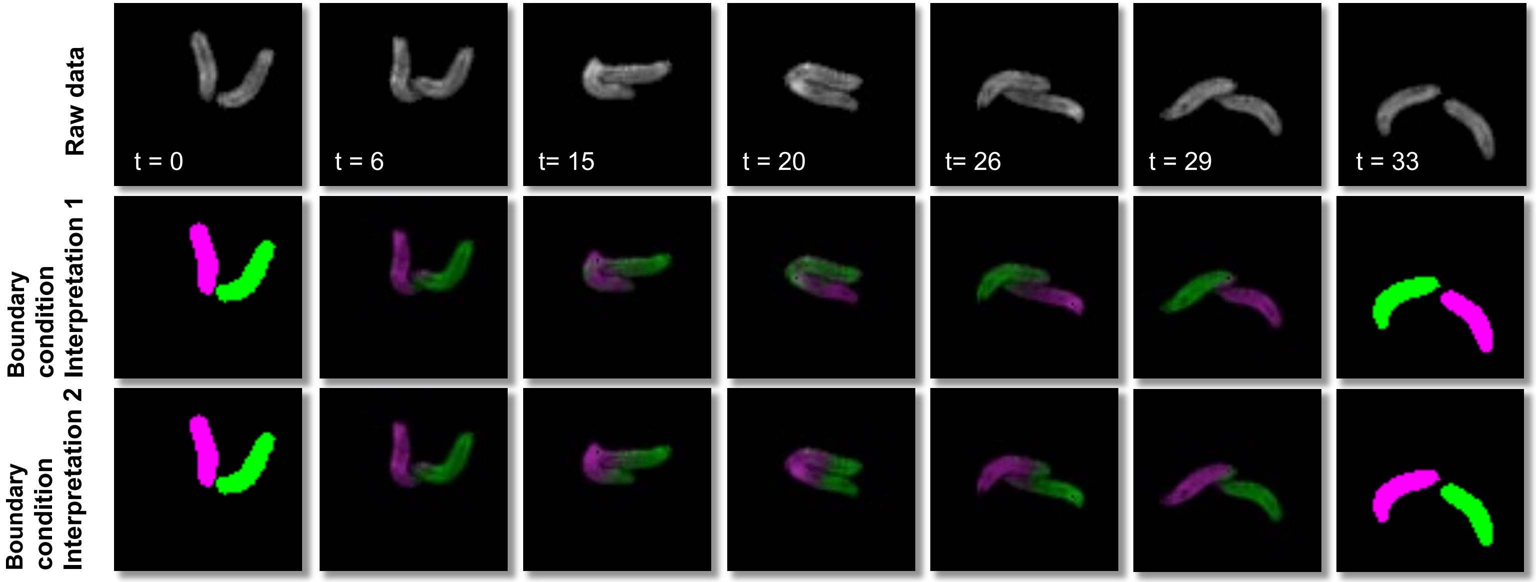

Raw data
Boundary condition Interpretation 1
Boundary condition Interpretation 2
t = 0
t = 6
t= 15
t = 20
t= 26
t = 29
t = 33

Figure 5.1: Top row: selected sub-frames from the raw data. Center and bottom rows: two solutions satisfying two interpretations of the sequence. The interpretations are encoded in terms of boundary conditions (here shown in saturated colors). Under our model (Eq. (5.1)), **inference** allows the estimation of the latent variable states (shown with non saturated colors) and gives the energy interpretation associated with the current boundary condition. Given training data, again in terms of boundary condition, **learning** finds model parameters so that the correct interpretation (central row) has lower energy than erroneous interpretations.

5.2 Related Work

Tracking multiple objects from a single camera view point has a long history in computer vision where most of the work has been focused on pedestrian and vehicle tracking. State of the art approaches which handle occlusion of such targets often leverage two key characteristics of the tracked objects: they are distinguishable, for example pedestrians with different clothes or cars of different colors, and/or they have smooth motion trajectories. These two characteristics have been firstly exploited in frame-by-frame tracking by combined motion and appearance models like in *Okuma et al.* (2004), and more recently by global data association techniques. These techniques achieve state of the art performance by integrating over time local appearance and motion cues in order to jointly optimize the identity assignments to candidate detections, e.g. *Nillius et al.* (2006); *Shitrit et al.* (2013); *Collins* (2012); *Zhang et al.* (2008); *Jiang et al.* (2007). In *Henriques et al.* (2011), Henriques *et al*. propose to use global inference in order to determine regions which results in merged detections and then match the objects before and after the ambiguous event using costs given by appearance and motion features.

The recent demand for automatic analysis of images in biology poses a new set of problems to the tracking community that are rarely found in natural images. In fact, objects of interest in biology are, on the one hand, highly deformable, often indistinguishable and sometimes translucent; and on the other hand, they can exhibit motion that appears stochastic and they may divide into multiple parts. In biological image analysis, most of the research which explicitly handles overlapping and deformable objects has focused on frame-by-frame tracking of blob-like structures such as cells *Li et al.* (2008); *Bise et al.* (2009). In *Li et al.* (2008), merged cells are separated by combining level sets with a motion filter, while *Bise et al.* (2009) uses frame-by-frame partial contour matching. The limitation of these frame-by-frame approaches is that they are prone to drift and sensitive to errors in the segmentation. Excellent work exists on mice tracking *Branson and Belongie* (2005) using particle filters to keep track of the object contours. This approach is limited to represent affine transformations of a limited number of manually annotated shape templates in order to reduce computational complexity. Ants tracking has been addressed in *Khan et al.* (2006) using a particle filter and adult Drosophila fly tracking has been addressed in*Branson et al.* (2009) using a constant velocity model. A recent review of open-source worm like objects trackers is *Husson et al.* (2012). According to the authors, these softwares work in uncrowded situations and do not handle overlap. The tracks are simple terminated when occlusion take place and reinitialized afterwards.

The problem of separating overlapping worms has been addressed by *Wählby et al.* (2010) for static image segmentation. This approach relies on candidate segments produced by skeletonization and performs best for elongated objects which have a low probability to lie side by side. The recent work *Fiaschi et al.* (2013) tracks groups of worm-like objects and detects overlap events. However, like in the algorithms reviewed in *Husson et al.* (2012), their approach is unable to separate the overlapping objects and disambiguate the individual identities which are lost when mutual occlusion take place.

The methods in *Branson and Belongie* (2005); *Bise et al.* (2009); *Li et al.* (2008); *Nillius et al.* (2006); *Henriques et al.* (2011); *Zhang et al.* (2008); *Shitrit et al.* (2013); *Jiang et al.* (2007); *Okuma et al.* (2004); *Collins* (2012) require parameters which are set manually. Our approach builds on the intuitions of Lou *et al*. *Lou and Hamprecht* (2011, 2012) and learns its parameters from partial user annotations. *Lou and Hamprecht* (2011) firstly posed object tracking as a structured learning problem, and *Lou and Hamprecht* (2012) showed how this learning could be achieved from partial annotations. Structured learning from partial labels is closely related to structured learning with latent variables *Yu and Joachims* (2009) as missing labels can be interpreted as unobserved variables. Our application and energy formulation differ from *Lou and Hamprecht* (2011, 2012), where the authors use integer linear programming (ILP) for the tracking of divisible, non-occluding cells. Our model draws ideas from the literature on optimal transportation theory (*Schrijver*, 2003) and on network flows (*Ahuja et al.*, 1993; *Bertsekas*, 1998). A particular transportation problem, the Earth Mover Distance (EMD) *Rubner et al.* (2000), finds the minimum cost flow between two distributions of masses, one of which is seen as the source and the other as the sink. This distance has numerous applications in tracking, e.g. *Oron et al.* (2012); *Ren and Malik* (2007); *Wang and Guibas* (2012). Ren *et al*. in *Ren and Malik* (2007) were the first to introduce EMD for frame-by-frame single object tracking. In that work the authors propose to predict the next position of a single target by using EMD to model the observed flow of intensity. EMD offered robustness to non-rigid deformation of the object. Recently, Oron *et al*. *Oron et al.* (2012) showed that EMD costs can be updated online, and used the distance to determine the weights of an adaptive particle filter. *Wang and Guibas* (2012) first showed that the costs of EMD can be learned from training data, by giving the distance order of triplets of points. The authors also proposed a learning strategy similar to gradient descent structured supprt vector machines. In our approach, we extend these ideas by allowing multiple different masses (namely different colors). We thus jointly minimize the cost of the multiple colored flows in similar fashion to a minimum cost multi-commodity flow problem (see *Ahuja et al.* (1993); *Bertsekas* (1998) for an introduction). Minimum cost multi-commodity flow problems are used in operations research to model the shipment cost of multiple distinguishable commodities over a network. Unlike the minimum-cost problem for single flow that can be efficiently solved with graph-cut, the minimum-cost multi-commodity flow problem for multiple integer flows is a NP-hard problem. However, when integer solutions are not required these problems can solved in polynomial time by Linear Programming (LP) (*Vanderbei*, 2008; *Bisschop and Roelofs*, 2006). Our convex energy formulation, Eq. (5.1) can be rewritten as general convex multi-commodity flow problem. Convex multi-commodity flow problems are generalizations of the minimum cost multi-commodity flow problems where the energy function is allowed to be an arbitrary convex function (cf. *Bertsekas* (1998)).

5.3 Problem Formulation and Modeling

Our key idea is sketched in Fig. 5.1. When two indistinguishable objects overlap and then separate again, we have two possible interpretations of their identity assignment. To build intuition, assume that each object has a unique color but that we have only a monochrome sensor. In such a situation, the color of each pixel is a *latent* variable, while the pixel grey value intensities are *observed* data. Given a boundary condition determined by the current interpretation, our model is an energy function which allows estimating the state of all latent variables as the minimum energy solution. The aim of learning is to find model parameters such that the correct interpretation gets assigned the lowest energy.

Our model is a generalization of EMD in the sense that the motion of multiple larvae is modeled as the flow of multiple *differently* colored masses. These flows are optimized simultaneously across several frames, subject to costs that express the following notions:

- To obtain the most parsimonious interpretation of the action, colored masses should move around as little as possible, while still satisfying the boundary condition.
- The spatial distribution of each color should be smooth inside each object.
- The intensity of all colors in a pixel should sum up to approximate the observed overall intensity in that pixel.
- Conservation of colors over time holds only approximately, to account for overall fluctuations in the image intensity, as well as Poisson and sensor noise or sensor saturation.

In formulating our energy function, we make two design choices. First, by using a weighted sum of costs derived from multiple features, we make sure that the model is sufficiently expressive to allow assigning the lowest energy to the correct solution for each training sample. Second, we restrict our formulation to terms that are linear in the latent variables, thus allowing for efficient optimization.

5.3.1 Precise formulation

The energy depends on two kinds of variables: the *flows* $\{f\}$ and the *masses* $\{m\}$. If we index each pixel inside the spatiotemporal volume of the video as $i = 1,...,N_P$, a colored mass variable m_i^k represents the intensity mass associated with larva $k = 1,...,N_L$ at pixel i. Multiple masses indexed by k are therefore associated with each pixel indexed by i. Masses can be observable variables or latent variables of the problem. In particular, the masses of pixels inside an *isolated* object are observable variables because during the learning we ask the user to provide label strokes for these pixels. Conversely, pixels of overlapping objects are associated with *latent* mass variables. In cases where we need to distinguish between observable and latent masses, we use respectively symbols $\dot{m}_i^k$ and $\mathring{m}_i^k$: $\{\dot{m}\}\cap\{\mathring{m}\} = \emptyset$ and $\{\dot{m}\}\cup\{\mathring{m}\} = \{m\}$. We reserve the symbol m_i^0 to represent the measured grey value intensity of a pixel i in the raw grayscale image.

To model motion, we define flows on a directed graph connecting pixels in the video. The graph connectivity is built on a spatiotemporal neighborhood $\mathcal{N}^{st}$ connecting pixels in consecutive frames. The flow variables f_{ij}^k, assigned to each edge $ij \in \mathcal{N}^{st}$, represent the flow of the mass associated with larva k between pixel i at time t and pixel j at time $t+1$. Flows are always latent variables of the problem and maintain temporal coherence between the pixels in consecutive frames. The energy function, giving expression to the notions from the previous section, reads as:

$$
\begin{aligned}
E(\underbrace{\mathbf{f},\mathring{\mathbf{m}}}_{\text{latent variables}};\dot{\mathbf{m}},\mathbf{w}) = {} & \sum_{l=3}^{M}\left[\underbrace{\sum_{k=1}^{N_L}\sum_{ij\in\mathcal{N}^{st}} f_{ij}^k d_{ij}^l}_{\text{flow cost term}}\right] w^l \\
& + \left[\underbrace{\sum_{k=1}^{N_L}\sum_{ij\in\mathcal{N}^{s}} |m_i^k - m_j^k| d_{ij}^2}_{\text{smoothness term}}\right] w^2 \\
& + \left[\underbrace{\sum_{i=1}^{N_P}\left|\sum_{k=1}^{N_L} m_i^k - m_i^0\right| d_{ii}^1}_{\text{data fidelity term}}\right] w^1 \\
& + \left[\underbrace{\sum_{i=1}^{N_P}\sum_{k=1}^{N_L}\left|\sum_{j\in\mathcal{N}_i^{st}} f_{ij}^k - m_i^k\right|}_{\text{outgoing flow conservation term}}\right] w^1 \\
& + \left[\underbrace{\sum_{j=1}^{N_P}\sum_{k=1}^{N_L}\left|\sum_{i\in\mathcal{N}_j^{st}} f_{ij}^k - m_j^k\right|}_{\text{incoming flow conservation term}}\right] w^1
\end{aligned}
\tag{5.1}
$$

Symbol	Definition
t	Time index $t = \{1, ..., T\}$
i, j	Pixel indices, there are N_p pixels inside the spatiotemporal volume of the video: $i, j \in \{1, ..., N_p\}$
$\mathcal{N}^s$	spatial neighborhood relates pixels in the same frame
$\mathcal{N}^{st}$	spatiotemporal neighborhood relates pixels in consecutive frames
l	Feature index, $l \in \{1, ..., M\}$
k	Mass color index, $k \in \{1, ..., N_L\}$
f_{ij}^k	Flow associated with larva/color k between pixels $i, j \in \mathcal{N}^{st}$
d_{ij}^l	Cost of flow between pixels $i, j \in \mathcal{N}^{st}$ computed from the l^{th} feature.
w^l	Weight associated with the l^{th} feature
m_i^k	Mass associated with larva/color k in pixel i. Masses can be: $\dot{m}_i^k$ observable variables $\mathring{m}_i^k$ latent variables
m_i^0	Measured grey value intensity in pixel i

Table 5.1: Notation

The energy is a sum of terms weighted by w^l, $l = 1, ..., M$. These terms fall into four categories: from the top of Eq. (5.1), *flow cost terms* penalize flow of mass between pixels which are different according to the parameter d_{ij}^l dependent on the l^{th} image feature. The costs d_{ij}^l can take into account not only spatial distance, as expressed by various powers of the Euclidean distance, but also dissimilarities in local appearance, *etc*. (see section 5.3.4). The *smoothness term* favors spatially smooth solutions in which adjacent pixels have similar colors. For each frame, this term is defined on pair of pixels within the spatial neighborhood $\mathcal{N}^s$. The *data fidelity term* favors solutions where the sum of the colors associated with each pixels is close to the grey value intensity m_i^0. Finally, the *flow conservation terms* enforce temporal consistency of proximate pixels at neighboring time steps. In classical flow problems these terms are imposed as linear constraints while we include these terms in the energy to account for fluctuations in the intensity. Although not required by the formulation, we find it convenient to give the same weight w^1 to data fidelity and flow conservation terms.

Under the boundary condition given by observed mass variables $\dot{\mathbf{m}}$, the optimization of Eq. (5.1) allows estimating the masses for the latent variables $\mathring{\mathbf{m}}$. The energy optimization problem

$$\min_{\mathbf{f},\mathring{\mathbf{m}}\in\mathbb{R}_+^M} E(\mathbf{f},\mathring{\mathbf{m}}\,;\,\dot{\mathbf{m}},\mathbf{w}) \tag{5.2}$$

is linear in the weights $\mathbf{w}$ and in the flows $\mathbf{f}$. The minimization of Eq. (5.1) can be solved efficiently by linear programming when replacing the absolute values with auxiliary variables[1] (cf. *Bisschop and Roelofs* (2006)).

5.3.2 Inference: identity interpretation

Problem 5.2 assumes the masses $\dot{\mathbf{m}}$, associated with the pixels of the isolated individuals as observed boundary conditions. However, to disambiguate the identity of the objects we are interested in finding the best assignment for the observable variables $\dot{\mathbf{m}}$, which provides the best *valid* interpretation of the video. We define a valid interpretation of a spatiotemporal sequence as an assignment for the mass variables $\dot{\mathbf{m}}$ which respects two constraints: first, each isolated object has all its pixels labeled by masses of the same color and second, for each timestep all isolated objects have a unique color. If we define $\mathcal{I}$ as the space of such valid identity interpretations, then in order to find the lowest energy identity assignment of the objects, we solve the following optimization problem:

$$\operatorname*{argmin}_{\substack{\dot{\mathbf{m}}\in\mathcal{I} \\ \mathbf{f},\mathring{\mathbf{m}}\in\mathbb{R}_+^M}} E(\mathbf{f},\mathring{\mathbf{m}},\dot{\mathbf{m}}\,;\,\mathbf{w}) \tag{5.3}$$

Rather than incorporating the constraints on $\mathcal{I}$ explicitly, we exploit the structure of the problem: we simply solve repeatedly for each of the $N_L!$ distinct interpretations that are possible for the identity assignment of those N_L objects that aggregate into a cluster. Note that, in most cases, N_L is only one or two in our data, provided that we break the entire video into overlapping and non-overlapping spatiotemporal regions using the result from algorithm *Fiaschi et al.* (2013) which is described in Chapter 4. When four or more objects overlap in a cluster we use an approximate strategy described in section 5.4.2.

[1] The optimization of Eq. (5.1) is implemented using ILOG Cplex.

5.3.3 Parameter learning from partial annotations

The learning determines the optimal weights $\mathbf{w}$ in Eq. (5.1) so that the correct interpretation of the training data receives lower energy than all erroneous interpretations. We solve this energy parametrization problem via structural risk minimization *Lou and Hamprecht* (2012); *Tsochantaridis et al.* (2006); *Yu and Joachims* (2009). Our training examples, $n = 1, ..., N$ for the observable colored mass variables are label strokes where the user marks the identity of isolated larvae only before and after the ambiguous region where they overlap, as depicted in Fig. 5.2. Arguably, this annotation requires a minimal click effort for the user. A training example $(\dot{\mathbf{m}}_n, \mathring{\mathbf{m}}_n, \mathbf{f}_n, \mathbf{d}_n)$ is composed of observed and latent variables and by the features of the image region $\mathbf{d}_n$. The loss function of the learning problem penalizes positive differences between the energy of the correct interpretation and the lowest energy among any of the wrong interpretations. If we define $\bar{\mathcal{I}}_n = \mathcal{I}_n \backslash \dot{\mathbf{m}}_n$ as the set of consistent but wrong boundary conditions then, similarly to *Yu and Joachims* (2009); *Lou and Hamprecht* (2012), the loss function can be written as:

$$\begin{aligned} L(\mathbf{w}, \dot{\mathbf{m}}_n) \; &= \; \min_{\mathbf{f}, \mathring{\mathbf{m}} \in \mathbb{R}_+^M} E(\mathbf{f}, \mathring{\mathbf{m}} \; ; \; \dot{\mathbf{m}}_n, \mathbf{w}) \\ &- \; \min_{\substack{\dot{\mathbf{m}} \in \bar{\mathcal{I}}_n \\ \mathbf{f}, \mathring{\mathbf{m}} \in \mathbb{R}_+^M}} \left[E(\mathbf{f}, \mathring{\mathbf{m}}, \dot{\mathbf{m}} \; ; \; \mathbf{w})] - \Delta(\dot{\mathbf{m}}_n, \dot{\mathbf{m}}) \right] \end{aligned} \tag{5.4}$$

Here, $\Delta(\dot{\mathbf{m}}_n, \dot{\mathbf{m}})$ is the task loss function which depends only on the observable variables. In this study, we use the number of identity switches between the predicted interpretation and the gold standard from the training set. Note that our restriction to the set of valid interpretations $\bar{\mathcal{I}}_n$ (see section 5.3.2) makes $\Delta(\dot{\mathbf{m}}_n, \dot{\mathbf{m}})$ a structured loss function which is not additive over local cliques of variables. The model is trained by finding the optimal weights which minimize the regularized structural risk:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|_2^2 \; + \; \frac{C}{N} \sum_n |L(\mathbf{w}, \dot{\mathbf{m}}_n)|_+ \\ \text{s.t.} \\ &w^1, ..., w^M \geq 0 \end{aligned} \tag{5.5}$$

Here, $|\cdot|_+$ is the hinge function and the additional constraints ensure convexity of Eq. (5.1).

The loss in Eq. (5.4) is the difference of two convex functions (DC). Such DC problems *Tao and An* (1997) are NP-hard, but can be solved approximately by using the CCCP procedure *Yuille and Rangarajan* (2003). Therefore, it is possible to apply CCCP for the solution of problem 5.5 as first proposed in *Yu and Joachims* (2009). Briefly, this procedure alternates between estimating the most probable state for the latent variables and solving the structured SVM problem treating all variables as observed. We implement the CCCP procedure based on the n-slack *Joachims* *et al.* (2009) formulation of structured SVM[2]. Taken together, this allows us to learn the parameters of the model from very weak annotations and generic features. We observe convergence of the CCCP procedure in less than seven iterations on average.

5.3.4 Features

A number of expressive features d_{ij}^l ensure that the model in Eq. (5.1) can differentiate between the possible interpretations of the sequence. This is in contrast to previous work *Nillius et al.* (2006); *Henriques et al.* (2011); *Zhang et al.* (2008); *Shitrit et al.* (2013); *Jiang et al.* (2007); *Collins* (2012) whose cost functions were designed manually. All features are summarized in table 5.2. Beside traditional features such as the powers of the Euclidean distance computed from the spatial locations of two pixels, we propose a new set of features that captures the local spatiotemporal structure of the data. In particular, these features are derived from the intensity profile $\Phi_{ij}(s)$, $s \in [0,1]$. This is computed for pixel i located in frame t and pixel j located in frame $t+1$, along the line connecting the spatial locations of the two pixels. As in table 5.2, we collect the features in three groups: the first group are the features multiplying the smoothness and the data fidelity term, the second group are all the purely spatial costs and the third group includes the proposed features derived from $\Phi_{ij}(s)$. The learned weights of these features are shown in Fig. 5.3.

Feature	
d_{ii}^1	$\exp\left(-m_i^0/255\right)$
d_{ij}^2	constant
d_{ij}^3	Spatial Euclidean distance between pixels i, j
d_{ij}^4	Second power of d_{ij}^3
d_{ij}^5	Fourth power of d_{ij}^3
d_{ij}^6	$\lvert m_i^0 - m_j^0 \rvert$
d_{ij}^7	$\int_0^1 \Phi_{ij}(s)ds$
d_{ij}^8	$\max_{s\in[0,1]} \Phi_{ij}(s) - \min_{s\in[0,1]} \Phi_{ij}(s)$
d_{ij}^9	$\langle \Phi_{ij}(s) - \langle \Phi_{ij}(s) \rangle^2 \rangle$
d_{ij}^{10}	$\int_0^1 I_{\{\Phi_{ij}(s) \le 50\}} ds$

Table 5.2: Definition of used features. I is an indicator function.

[2] Our implementation builds on the excellent freeware library https://github.com/amueller/pystruct

5.4 Experiments

In our experiments, we compare the proposed method to the closely related work *Branson et al.* (2009); *Lou and Hamprecht* (2011); *Henriques et al.* (2011).

On the one hand, we use the established software Ctrax *Branson et al.* (2009) that can track multiples *Drosophila* adults. Ctrax uses frame-by-frame tracking based on a constant velocity model and also incorporates automatic corrections for merged detections by considering multiple splitting hypotheses. On the other hand, we propose two methods inspired by the matching approach from *Henriques et al.* (2011) as baseline. Briefly, we compute the minimum cost matching between the isolated larvae entering the encounter region and those exiting the encounter region. Baseline 1 computes matching costs *only* based on differences in size of the larvae. This baseline tests our assertion that the larvae cannot be distinguished against the natural variability in size of the population. Baseline 2 combines the approach from *Henriques et al.* (2011) with *Lou and Hamprecht* (2011). Instead of using only one feature, as in baseline 1, we use multiple costs computed as linear combination of three features: Euclidean distance between the centre of the isolated larvae, difference in size and difference in average intensity. We then learn the weights with a structured SVM as proposed in *Lou and Hamprecht* (2011) using the same number of examples as in the proposed approach. Note that, in contrast to baseline 1, this baseline *combines* appearance and location information.

5.4.1 Dataset and evaluation metric

We have evaluated our approach on a challenging dataset of larvae tracking composed of 33 high resolution movies[3]. Each movie has a length of 5 minutes, a temporal resolution of 3.3 frames per second (1000 frames in total) and contains on average 20 larvae. The spatial resolution is 135.3 μm/pixel at 1400×1400 pixels/image. For the preprocessing and the construction of the counting graph we follow the guidelines from *Fiaschi et al.* (2013). That counting algorithm obtained a precision of 99.9% on this dataset, providing an almost perfect tracking for the isolated larvae. We extract all subregions containing encounters of two or more larvae from the counting graph. A human expert manually labeled each larva entering or leaving each region in order to create a gold standard, as sketched in Fig. 5.2. Similarly to *Keni and Rainer* (2008), performance is measured by counting the number of identity mismatches between the output of the algorithm and the gold standard. Missed detections or lost tracks, as result from other algorithms, are added to the error. This metric is then normalized by the total number of objects entering the region.

[3] upon request luca.fiaschi@iwr.uni-heidelberg.de

Method	Total	enc=2	enc=3	enc$\geq$ 4
Baseline 1	17.1%	15.6%	26.3%	67.8%
Baseline 2	11.8%	10.5%	23.1%	42.8%
Ctrax *Branson et al.* (2009)	13.2%	11.5%	26.3%	57.1%
Our proposal	**5.3**%	4.2%	**14.7**%	**32.1**%

Table 5.3: Task loss across the dataset: weighted average total and breakdown into encounters of two, three or four and more larvae.

5.4.2 Implementation and experimental details

Three approximations were made in order to reduce the computational effort of the proposed method. First, for each encounter we select up to 15 sub-frames linearly spaced in time in order to reduce the number of variables involved in the optimization. Second, the threshold for the spatiotemporal neighborhood $\mathcal{N}^{st}$ is chosen for each encounter as the minimum distance such that each foreground pixel is connected by at least one temporal edge. On average, the temporal neighborhood considers a radius of 10 pixels while the spatial neighborhood $\mathcal{N}^{s}$ is fixed to a radius of 1 pixel. Third, when four or more larvae are in the overlapping region, baseline 2 is used to retrieve only the first six interpretations with lowest matching cost. Among these, with our main method, we select the interpretation which obtains the lowest energy.

Our model as well as baseline 2 are trained using only 25 examples of encounters of two larvae (3% of all available encounters) and testing is performed on the entire dataset. Most encounters have a very short duration, so to harvest more hard examples, we perform a first inference round and then train the model again with a mixture of 15 randomly selected and 10 hard examples. To asses the performance with a randomly selected training set, the learning curve of the algorithm with different number of training examples is included in the supplementary material. Training our model takes around 8 hrs and inference around 1 day on a 32 cores 2.4 GHz Intel Quad-Xeon machine.

5.4.3 Results

Fig. 5.4 and Fig. 5.5 illustrate our results. For two difficult cases from the test set, Fig. 5.4 shows the inferred state of the latent masses of the interpretation with lowest energy. Fig. 5.5 compares the six possible interpretations of an encounter of three larvae.

The quantitative comparison between our approach and the other methods is presented in table 5.3. The two baseline methods and `Ctrax` perform similarly on this dataset. Baseline 1, which exploits only size, has the worst performance while baseline 2, which uses combined appearance and position, performs slightly better than `Ctrax`. It learns its parameters from training examples. For encounters of two or three larvae, our proposal produces consistently more accurate results without using any appearance feature. On encounters of four or more larvae, where baseline 2 is used to retrieve candidate interpretations, our approach can improve the results without however reaching truly satisfactory performance. For all methods, performance decreases with the number of larvae in the encounter as the scene becomes more cluttered and the number of interpretations increases. Given the difficulty of much of the data, the error rate achieved for encounters of two larvae is quite respectable. Even so, for agglomerates of more individuals, the resulting error is still high and, in those settings, computation times also need to be improved further.

5.5 Discussion

In this chapter we have presented and evaluated a new tracking algorithm that specifically addresses the identity switching problem for the challenging situation of multiple indistinguishable overlapping objects. The model builds on the assumptions that we have an unknown number of possibly indistinguishable and self-occluding but at least partially translucent objects. These objects are allowed to be deformable and perform complex motions, such as crawling across each other. All in all, Eq. (5.1) expresses a “least action principle" – after appropriate parametrization, a scene can be interpreted in terms of a minimal-cost transformation, or transport of masses of different colors. The structured learning framework allows to fine-tune the costs for these transportation processes so as to make the true solutions, as given by the training set, the least costly. A training set can be compiled very conveniently by specifying the identity of individual larvae only when they enter or leave an agglomerate. We solve the associated hard latent variable learning problem, and achieve encouraging results on challenging sequences of social larvae. While the computational complexity of our approach does not yet allow tackling large populations, this algorithm provides behavioural biologist a way of studying larvae contacts in low density populations (≈ 20 animals).

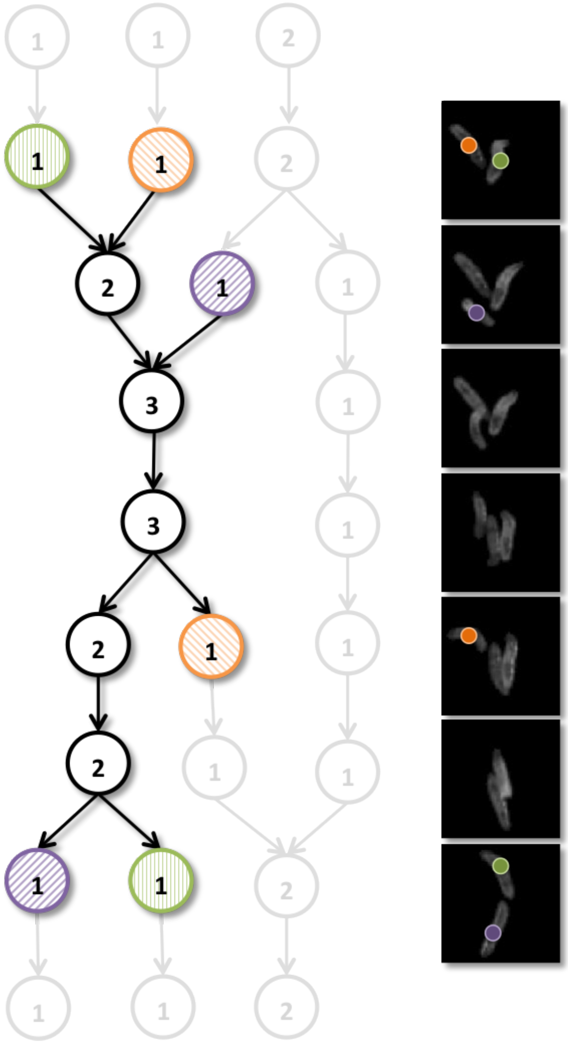


Figure 5.2: Representation of a training example corresponding to the occluded spatiotemporal region. On the left side, the counting graph as obtained by the algorithm of *Fiaschi et al.* (2013): each connected component of the foreground is depicted with a circle labeled with the number of objects it contains. Bold lines indicate a subgraph form the counting graph of *Fiaschi et al.* (2013) where an encounter of 3 larvae is detected. On the right side, each frame shows *only* the larvae that are in the region of interest defined by the subgraph on the left side. On the left and right sides: the colors indicate the identity of the isolated objects which enter or leave the region of interest, as labeled by the user during the training phase.

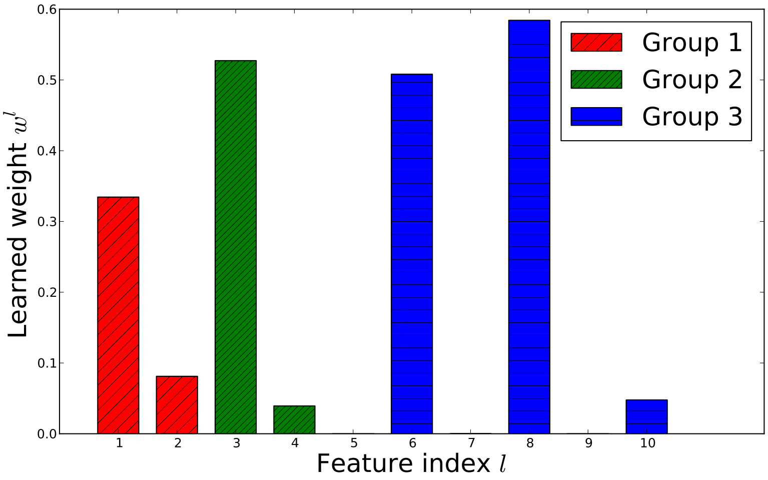


Figure 5.3: : Parameters $\mathbf{w}$ learned from the training data and normalized to unit norm, i.e. $\|\mathbf{w}\|_2 = 1$.

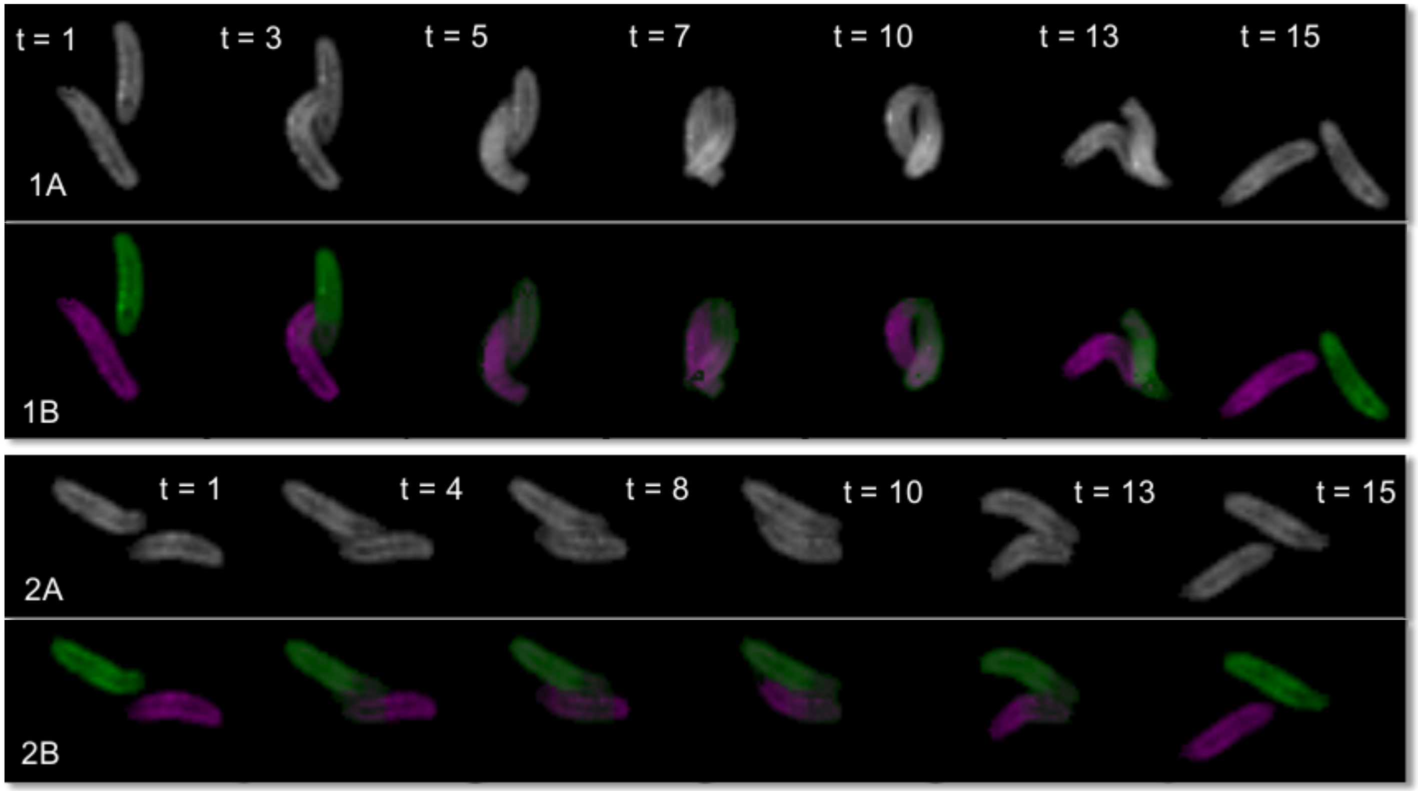


Figure 5.4: Two difficult examples from the test set. Raw data and lowest energy interpretation that is inferred by our algorithm. In the first (t=1) and the last frame (t=15) of each sequence the objects are separated and the value for the masses are set by the boundary conditions. In the intermediate frames of the sequence ($1 < t < 15$) the sum of the two colors is close to the grayvalue intensity in the raw data.

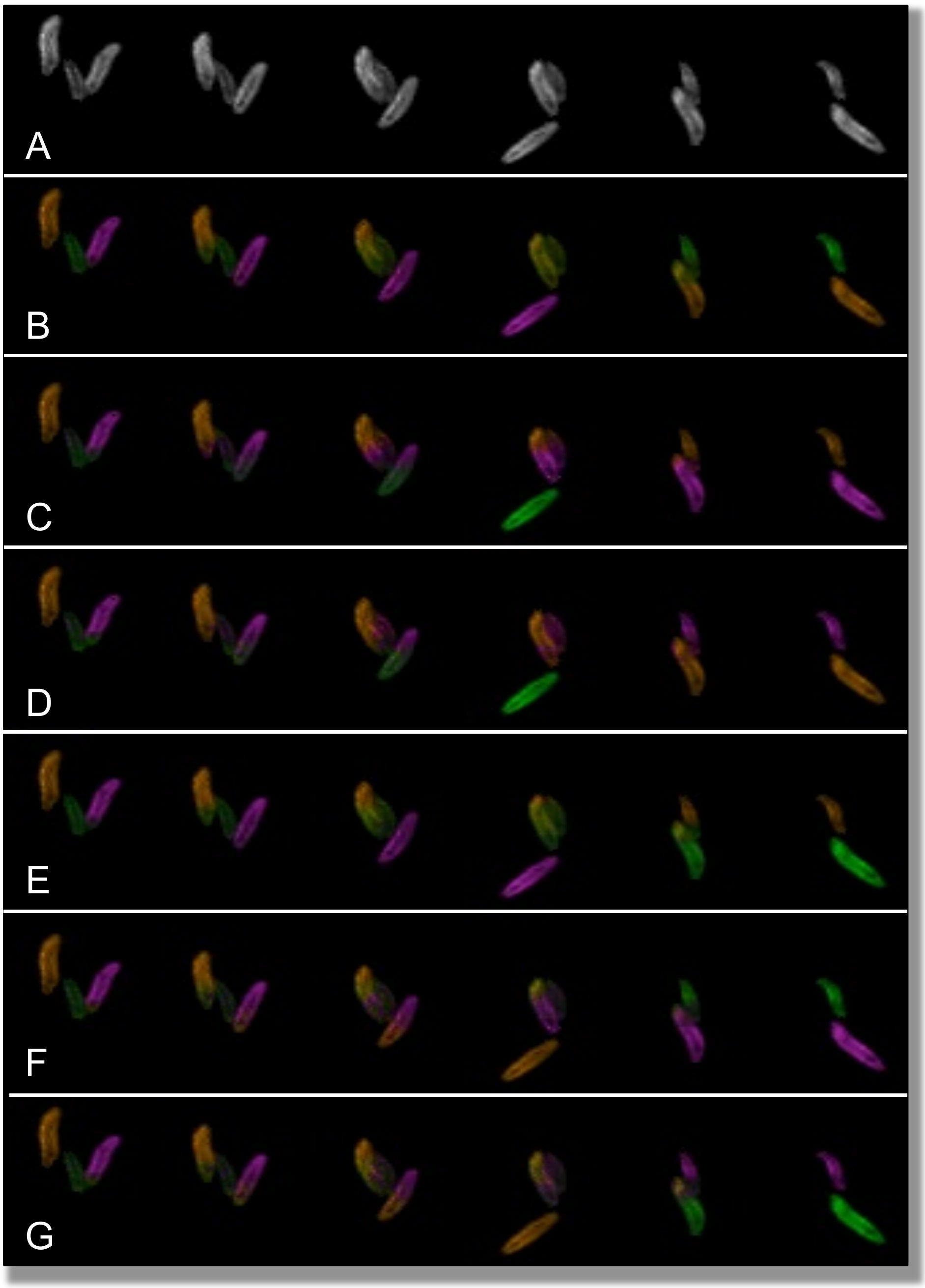


Figure 5.5: Selected sub-frames from an encounter of 3 larvae. From the top: raw data (A) and the 6 possible interpretations ranked by increasing energy. Our apporach correctly assigns the lowest energy to the interpretation in the second row (B).

Chapter 6

Conclusion

6.1 Conclusions and main contributions

This thesis aims to find answers to concrete challenges in biological image analysis. Our first contribution is to propose concrete algorithms to solve four relevant applications in the areas of *large scale data analysis* and *multiple object tracking*. The bottom line of our work is the investigation of learning based approaches to provide life scientists with the flexibility to adapt to their data. Chapter 2 presents a weakly supervised algorithm for detecting defects in HCS microscopic images. Indeed, quality control is often the first step of a careful automated data analysis workflow. Counting cells in microscopic images has been addressed in Chapter 3, where a density-counting algorithm based on supervised learning of spatially local structured predictors is introduced. Chapter 4 shows that object counting can be improved by exploiting temporal consistence in video sequences. The approach of Chapter 4 becomes a fundamental building block in Chapter 5, where we introduce a novel tracking algorithm for multiple objects. Again, our proposal is based on a weakly supervised learning strategy (structured learning from partial annotations). Besides improving over a yet largely unsolved problem in computer vision (tracking multiple object under heavy mutual occlusion), the latter proposal finds its motivation and application in the scientific quest to clarify the social behaviour of an important model organism such as Drosophila.

All the methods presented in this work are validated through extensive experiments conducted on a large variety of datasets and have demonstrated their accuracy and robustness. Also, for each method together with the advantages and the results we have discussed its limitations. This scientific attitude has inspired and connected the research directions presented in this thesis. For example, density-counting as presented in Chapter 3 is appropriate to count many objects with similar appearance even in presence of strong overlap. The counts are accurate when integrated over the entire image but the method fails estimating counts for small regions. This limitation has motivated the algorithm in Chapter 4, that produces more accurate results leveraging temporal consistency in video sequences.

The main methods that have been presented in this thesis build on the cutting edge computational approaches. As reviewed in Chapter 1 and within each chapter introduction, we draw ideas from the most recent research in machine learning, computer vision and optimization, e.g. learning structured models from partial annotations (Chapter 5). While sitting on the shoulder of the giants each chapter also introduces novelty in a concrete application domain. At the same time, we aim at demonstrating the effectiveness of a general modeling methodology: the design of structured models in conjunction with learning methods. Structured models, and in particular MRFs, are a powerful language to leverage prior information (e.g. neighboring pixels should have similar labels), while learning from (possibly weak) annotations avoids tedious parameter tuning and helps adapting a workflow to different datasets (under similar experimental conditions). We motivate these claims by reviewing the most significant contributions introduced in this thesis:

- **Object detection:** A popular framework for object detection uses *two classes* supervised classification in combination with a sliding window to locate an object in the image. In Chapter 2, we propose a weakly supervised algorithm for detecting defects in microscopic HCS. Our algorithm uses one class-learning and casts the problem as a novelty detection task. To the best of our knowledge, it is the first approach that performs detection (place local defects in the images) with a cascade of weakly supervised *one-class* classifiers. Unlike the previous work, this approach *does not* require labels for the objects to be detected.

- **Object counting in static images:** As reviewed in Chapter 3, the state of the art approach in object counting for static images is the density-counting technique (*Lempitsky and Zisserman*, 2010; *Fiaschi et al.*, 2012). In previous work, the counts for neighboring pixels were independent which leaded to spatially rough predictions. In Chapter 3 we propose to use a local structured predictor over small overlapping patches. Our approach is conceptually simpler than the previously used method while obtaining an overall performance similar to the state of the art. Even if our training procedure minimizes only the local pixel error on the density, which is an upper bound on the objects counting error, the locally structured predictors in combination with a strong nonlinear classifier (Random Forests) obtains good accuracy on the object counts.

- **Object counting in videos:** When analysing video sequences, previous objects counting approaches did not fully leverage the temporal domain. The reason is that the features that are mainly used in the literature have limited spatial and temporal support. In Chapter 4 we introduce a structured model (deterministic graphical model) to fuse multiple local predictions and obtain a globally consistent inference across all frames of the video sequence. We show that this structured model provides very accurate results outperforming predictions that are local in space and time.

- **Occlusion detection in videos:** As we reviewed in Chapter 4 occlusion detection has been exploited by tracking methods. The algorithm presented in Chapter 4 results in very accurate detection of mutual occlusion events. Given this result, in Chapter 5 we decompose a large tracking problem into ambiguous and unambiguous spatiotemporal regions. Unambiguous regions are easy to track and correspond to the path of a single animal between two occlusion events. Ambiguous regions are sparse but difficult to track. They correspond to mutual visual occlusion of the targets. The ambiguous regions are then analysed by a more powerful, but also more computationally expensive, tracking algorithm which includes an explicit modelling of the occlusion dynamics (Chapter 5).

- **Multiple drosophila larvae tracking:** As extensively reviewed in Chapter 5, most of the tracking approaches for worm shaped targets are non robust to mutual occlusion and prone to identity switching errors. This currently hinders the possibility to study the social behaviour of Drosophila larvae. Chapter 5 proposes an algorithm for multiple larvae tracking in a low density population. To the best of our knowledge, this algorithm, for the first time, explicitly minimizes (in a max margin sense) identity switches errors when objects cross each other.

- **Structured models and structured learning:** most tracking methods rely on manual tuning of several hyper-parameters, also in settings which can be challenging even to humans: tracking multiple indistinguishable and overlapping objects (Chapter 5), we have chosen a n approach that has demonstrated, once again, the effectiveness of structured learning. We *first design* a richly parametrized energy function that models, for the first time, the mutual occlusion of translucent and indistinguishable objects. Our model encodes the visual observation that the scene resembles the mixing of multiple fluids. We *then learn* the correct parametrization of the energy from partially annotated training data. During training, our approach only requires the user to indicate the identity of objects by placing marks before and after occlusion events.

- **Generation of datasets:** Through the development of this thesis, a large benchmark dataset for multiple larvae tracking has been compiled and will be made available to the community[1]. This dataset includes a long sequence (1000 frames) with high spatial and temporal resolution and provides the track for each animal in a large population of around 300 individuals.

[1] At the moment available upon request: `luca.fiaschi@iwr.uni-heidelberg.de`

- **Scientific open-source software** We are currently making a great effort in providing the methods that we have developed as a concrete support for researchers in biology and medicine. The code for the experiments in Chapter 3 and Chapter 4 are publicly available[2]. Moreover, given the large applicability of the counting algorithm presented in Chapter 3, we are implementing a modular and easy-to-use workflow in the open source software ILASTIK[3].

In summary, to address some of the challenges of modern biological image analysis we have investigated and proposed many innovative and original solutions based on machine learning. Our main contribution is to demonstrate the possibility and the advantages of learning from weak and partial annotations on real world data and on challenging and relevant applications. Given the rapid development of microscopy techniques, we foresee that these two concepts will become more and more important and will find applications and extensions in many more scenarios.

[2] https://github.com/lfiaschi

[3] www.ilastik.org

6.2 Future research directions

In the future, we would like to further contribute to the scientific research by extending our work in the following directions:

- **Interactive training of learning to count:** Acquiring training examples is the current bottleneck of all learning based counting approaches which are reviewed in Chapter 3 and 4, including our proposal. These methods require fully annotated images. Annotations can become very time consuming as each image can contain hundreds or thousands of objects (e.g. pedestrians in a public square during a sit-in). In such a situation, an interactive algorithm can tremendously speedup the training procedure.

- **Convex optimization:** We would like to scale the tracking approach proposed in Chapter 5 to large populations comprising hundreds of larvae. Running time is the current bottleneck of our proposal. The formulation Eq. (5.1) proposed Chapter 5 is a convex multi-commodity flow problem. Decomposition algorithms, such as Dantzig-Wolfe decomposition (*Dantzig and Wolfe*, 1960), have been successfully applied to large scale multi-commodity problems (*Rios and Ross*, 2010). A proper decomposition algorithm for Eq. (5.1) could speedup computation and allow tracking in larger animal populations.

- **Shape prior for combined tracking and segmentation:** The occlusion model proposed in Chapter 5, which is formulated as an energy function, does not model explicitly the shape of the tracked larvae. RBMs(*Smolensky*, 1986) are an instance of energy based models which can capture high order correlation among random variables. RBMs have been shown to be a powerful representation of object shapes (*Eslami et al.*, 2012) and have been used in single object segmentation (*Chen et al.*, 2013; *Kae et al.*, 2013). In the future, the inclusion of an energy term to model the larvae shape (shape prior) could allow joint segmentation and tracking of the animals.

Bibliography

Adami, C., 2002. What is complexity? BioEssays 24 (12), 1085–1094.

Ahuja, R. K., Magnanti, T. L., Orlin, J. B., 1993. Network flows: theory, algorithms, and applications. Prentice Hall.

Amit, Y., Geman, D., 1997. Shape quantization and recognition with randomized trees. Neural computation 9 (7), 1545–1588.

Andres, B., Kroeger, T., Briggman, K. L., Denk, W., Korogod, N., Knott, G., Koethe, U., Hamprecht, F. A., 2012. Globally optimal closed-surface segmentation for connectomics. In: ECCV.

Andriluka, M., Roth, S., Schiele, B., 2010. Monocular 3d pose estimation and tracking by detection. In: CVPR.

Bakir, G., Hofmann, T., Schölkopf, B., Smola, A., Taskar, B., Vishwanathan, S., 2007. Predicting Structured Data. Advances in neural information processing systems. The MIT Press.

Bertsekas, D. P., 1998. Network optimization: continuous and discrete models. Athena Scientific Belmont.

Bigun, J., Granlund, G. H., 1987. Optimal orientation detection of linear symmetry. In: First International Conf. Comput. Vision.

Bise, R., Li, K., Eom, S., Kanade, T., 2009. Reliably tracking partially overlapping neural stem cells in dic microscopy image sequences. In: MICCAI Workshop on OPTIMHisE.

Bise, R., Yin, Z., Kanade, T., 2011. Reliable cell tracking by global data association. In: ISBI.

Bishop, C. M., Nasrabadi, N. M., 2006. Pattern recognition and machine learning. New York: Springer.

Bisschop, J., Roelofs, M., 2006. AIMMS - The Users's Guide. Paragon Decision Technology.

Blake, A., Kohli, P., Rother, C., 2011. Markov random fields for vision and image processing. The MIT Press.

Blaschko, M. B., Lampert, C. H., 2008. Learning to localize objects with structured output regression. In: ECCV.

Born, M., Wolf, E., 1999. Principles of optics: electromagnetic theory of propagation, interference and diffraction of light. Pergamon Press, New York.

Bosch, A., Zisserman, A., Muoz, X., 2007. Image classification using random forests and ferns. In: ICCV.

Bousquet, O., Bottou, L., 2007. The tradeoffs of large scale learning. In: Advances in Neural Information Processing Systems.

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

Branson, K., Belongie, S., 2005. Tracking multiple mouse contours without too many samples. In: CVPR.

Branson, K., Robie, A. A., Bender, J., Perona, P., Dickinson, M. H., 2009. High-throughput ethomics in large groups of drosophila. Nature methods 6 (6), 451–457.

Bray, M.-A., Fraser, A. N., Hasaka, T. P., Carpenter, A. E., 2012. Workflow and metrics for image quality control in large-scale high-content screens. Journal of biomolecular screening 17 (2), 266–274.

Breiman, L., 2001. Random forests. Machine learning 45 (1), 5–32.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., Sander, J., 2000. Lof: identifying density-based local outliers. In: ACM Sigmod Record. Vol. 29. ACM, pp. 93–104.

Cardona, A., Tomancak, P., 2012. Current challenges in open-source bioimage informatics. Nature methods 9 (7), 661–665.

Carpenter, A. E., Jones, T. R., Lamprecht, M. R., Clarke, C., Kang, I. H., Friman, O., Guertin, D. A., Chang, J. H., Lindquist, R. A., Moffat, J., et al., 2006. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. Genome biology 7 (10), R100.

Chan, A., Liang, Z., Vasconcelos, N., 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking. In: CVPR.

Chandola, V., Banerjee, A., Kumar, V., 2007. Outlier detection: A survey. ACM Computing Surveys, to appear.

Chen, F., Yu, H., Hu, R., Zeng, X., 2013. Deep learning shape priors for object segmentation. In: CVPR.

Cho, S., Chow, T., Leung, C., 1999. A neural-based crowd estimation by hybrid global learning algorithm. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 29 (4), 535–541.

Chung, K., Wallace, J., Kim, S.-Y., Kalyanasundaram, S., Andalman, A. S., Davidson, T. J., Mirzabekov, J. J., Zalocusky, K. A., Mattis, J., Denisin, A. K., et al., 2013. Structural and molecular interrogation of intact biological systems. Nature 497 (7449), 332–337.

Ciresan, D., Giusti, A., Schmidhuber, J., et al., 2012. Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in Neural Information Processing Systems 25. pp. 2852–2860.

Ciresan, D. C., Giusti, A., Gambardella, L. M., Schmidhuber, J., 2013. Mitosis detection in breast cancer histology images with deep neural networks. In: MICCAI.

Collins, R. T., 2012. Multitarget data association with higher-order motion models. In: CVPR.

Criminisi, A., Shotton, J., Konukoglu, E., 2011. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Microsoft Research technical report, 1–151.

Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C., 2004. Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision, ECCV. Vol. 1. p. 22.

Culotta, A., McCallum, A., 2005. Reducing labeling effort for structured prediction tasks. In: Proceedings of the National Conference on Artificial Intelligence (AAAI).

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: CVPR.

Dantzig, G. B., Wolfe, P., 1960. Decomposition principle for linear programs. Operations research 8 (1), 101–111.

Danuser, G., 2011. Computer vision in cell biology. Cell 147 (5), 973–978.

Echeverri, C. J., Perrimon, N., 2006. High-throughput rnai screening in cultured cells: a user's guide. Nature Reviews Genetics 7 (5), 373–384.

Eslami, S. A., Heess, N., Winn, J., 2012. The shape boltzmann machine: a strong model of object shape. In: CVPR.

Fiaschi, L., Konstantin, G., Afonso, B., Zlatic, M., Hamprecht, F. A., 2013. Keeping count: leveraging temporal context to count heavily overlapping objects. In: ISBI.

Fiaschi, L., Nair, R., Koethe, U., Hamprecht, F., 2012. Learning to count with regression forest and structured labels. In: ICPR.

Fontaine, E., Barr, A., Burdick, J., 2007. Model-based tracking of multiple worms and fish. In: ICCV Workshop on Dynamical Vision.

Gall, J., Yao, A., Razavi, N., Van Gool, L., Lempitsky, V., 2011. Hough forests for object detection, tracking, and action recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33 (11), 2188–2202.

Godec, M., Roth, P. M., Bischof, H., 2012. Hough-based tracking of non-rigid objects. Computer Vision and Image Understanding.

Goode, A., Sukthankar, R., Mummert, L., Chen, M., Saltzman, J., Ross, D., Szymanski, S., Tarachandani, A., Satyanarayanan, M., 2008. Distributed online anomaly detection in high-content screening. In: ISBI.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182.

Hastie, T., Tibshirani, R., Friedman, J. J. H., 2001. The elements of statistical learning. Springer New York.

Held, M., Schmitz, M. H., Fischer, B., Walter, T., Neumann, B., Olma, M. H., Peter, M., Ellenberg, J., Gerlich, D. W., 2010. Cellcognition: time-resolved phenotype annotation in high-throughput live cell imaging. Nature methods 7 (9), 747–754.

Henriques, J., Caseiro, R., Batista, J., 2011. Globally optimal solution to multi-object tracking with merged measurements. In: ICCV.

Hero, A. O., 2006. Geometric entropy minimization (gem) for anomaly detection and localization. In: NIPS.

Hey, A. J., Tansley, S., Tolle, K. M., et al., 2009. The fourth paradigm: data-intensive scientific discovery. Microsoft Research Redmond, WA.

Husson, S. J., Costa, W. S., Schmitt, C., Gottschalk, A., et al., 2012. Keeping track of worm trackers. WormBook: the online review of C. elegans biology, 1–17.

Idrees, H., Saleemi, I., Seibert, C., Shah, M., 2013. Multi-source multi-scale counting in extremely dense crowd images. CVPR.

Jähne, B., 2002. Digital image processing. Measurement Science and Technology 13 (9), 1503.

Jain, A. K., Farrokhnia, F., 1991. Unsupervised texture segmentation using gabor filters. Pattern recognition 24 (12), 1167–1186.

Jain, V., Seung, H. S., Turaga, S. C., 2010. Machines that learn to segment images: a crucial technology for connectomics. Current opinion in neurobiology 20 (5), 653–666.

Jiang, H., Fels, S., Little, J. J., 2007. A linear programming approach for multiple object tracking. In: CVPR.

Joachims, T., 1999. Making large scale svm learning practical.

Joachims, T., Finley, T., Yu, C.-N. J., 2009. Cutting-plane training of structural svms. Machine Learning 77 (1), 27–59.

Jordan, M. I., 2004. Learning in graphical models. The MIT Press.

Kae, A., Sohn, K., Lee, H., Learned-Miller, E., 2013. Augmenting crfs with boltzmann machine shape priors for image labeling. In: CVPR.

Kappes, J. H., Andres, B., Hamprecht, F. A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B. X., Lellmann, J., Komodakis, N., et al., 2013. A comparative study of modern inference techniques for discrete energy minimization problems. In: CVPR.

Kausler, B., Schiegg, M., Andres, B., Lindner, M., Koethe, U., Leitte, H., Wittbrodt, J., Hufnagel, L., Hamprecht, F., 2012. A discrete chain graph model for 3d+ t cell tracking with high misdetection robustness. In: ECCV.

Kaynig, V., Fischer, B., Buhmann, J. M., 2008. Probabilistic image registration and anomaly detection by nonlinear warping. In: CVPR.

Keller, P. J., Schmidt, A. D., Wittbrodt, J., Stelzer, E. H., 2008. Reconstruction of zebrafish early embryonic development by scanned light sheet microscopy. Science 322 (5904), 1065–1069.

Keni, B., Rainer, S., 2008. Evaluating multiple object tracking performance: the clear mot metrics. EURASIP Journal on Image and Video Processing.

Khan, Z., Balch, T., Dellaert, F., 2006. Mcmc data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. TPAMI 28 (12), 1960–1972.

Knorr, E. M., Ng, R. T., 1998. Algorithms for mining distance-based outliers in large datasets. In: Proceedings of the 24rd International Conference on Very Large Data Bases.

Kollar, D., Friedman, N., 2009. Probabilistic graphical models: principles and techniques. The MIT Press.

Komodakis, N., Paragios, N., Tziritas, G., 2007. Mrf optimization via dual decomposition: Message-passing revisited. In: ICCV.

Komodakis, N., Tziritas, G., 2007. Approximate labeling via graph cuts based on linear programming. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (8), 1436–1453.

Kong, D., Gray, D., Tao, H., 2006. A viewpoint invariant approach for crowd counting. In: ICPR.

Kontschieder, P., Bulo, S. R., Bischof, H., Pelillo, M., 2011. Structured class-labels in random forests for semantic image labelling. In: ICCV.

Krähenbühl, P., Koltun, V., 2011. Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS.

Kreshuk, A., Straehle, C. N., Sommer, C., Koethe, U., Cantoni, M., Knott, G., Hamprecht, F. A., 2011. Automated detection and segmentation of synaptic contacts in nearly isotropic serial electron microscopy images. PloS one 6 (10), e24899.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. In: NIPS.

Kroeger, T., Mikula, S., Denk, W., Koethe, U., Hamprecht, F. A., 2013. Learning to segment neurons with non-local quality measures. In: MICCAI.

Lazarevic, A., Kumar, V., 2005. Feature bagging for outlier detection. In: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. ACM.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86 (11), 2278–2324.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F., 2006. A tutorial on energy-based learning. Predicting Structured Data.

Lempitsky, V., Zisserman, A., 2010. Learning to count objects in images. In: NIPS.

Li, K., Chen, M., Kanade, T., Miller, E. D., Weiss, L. E., Campbell, P. G., 2008. Cell population tracking and lineage construction with spatiotemporal context. Medical image analysis 12 (5), 546.

Liu, F. T., Ting, K. M., Zhou, Z.-H., 2008a. Isolation forest. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on.

Liu, R., Li, Z., Jia, J., 2008b. Image partial blur detection and classification. In: CVPR.

Lou, X., Fiaschi, L., Koethe, U., Hamprecht, F. A., 2012. Quality classification of microscopic imagery with weakly supervised learning. In: MICCAI-MLMI Workshop.

Lou, X., Hamprecht, F. A., 2011. Structured learning for cell tracking. In: NIPS.

Lou, X., Hamprecht, F. A., 2012. Structured learning from partial annotations. In: ICML.

Lowe, D. G., 1999. Object recognition from local scale-invariant features. In: CVPR.

Lucchi, A., Li, Y., Smith, K., Fua, P., 2012. Structured image segmentation using kernelized features. In: ECCV.

Ma, Z., Chan, A. B., 2013. Crossing the line: Crowd counting by integer programming with local features. CVPR.

Marana, A., Velastin, S., Costa, L., Lotufo, R., 1997. Estimation of crowd density using image processing. In: Image Processing for Security Applications (Digest No.: 1997/074), IEE Colloquium on. IET, pp. 11–1.

Mateescu, R., Dechter, R., 2008. Mixed deterministic and probabilistic networks. Annals of Mathematics and Artificial Intelligence 54 (1), 3–51.

Matov, A., Applegate, K., Kumar, P., Thoma, C., Krek, W., Danuser, G., Wittmann, T., 2010. Analysis of microtubule dynamic instability using a plus-end growth marker. Nature methods 7 (9), 761–768.

Megason, S. G., Fraser, S. E., 2007. Imaging in systems biology. Cell 130 (5), 784–795.

Meinshausen, N., 2006. Quantile regression forests. The Journal of Machine Learning Research 7, 983–999.

Mersch, D. P., Crespi, A., Keller, L., 2013. Tracking individuals shows spatial fidelity is a key regulator of ant social organization. Science 340 (6136), 1090–1093.

Micheva, K. D., Smith, S. J., 2007. Array tomography: a new tool for imaging the molecular architecture and ultrastructure of neural circuits. Neuron 55 (1), 25–36.

Mnih, V., Larochelle, H., Hinton, G. E., 2012. Conditional restricted boltzmann machines for structured output prediction. arXiv preprint arXiv:1202.3748.

Montillo, A., Shotton, J., Winn, J., Iglesias, J. E., Metaxas, D., Criminisi, A., 2011. Entangled decision forests and their application for semantic segmentation of ct images. In: Information Processing in Medical Imaging. Springer, pp. 184–196.

Moosmann, F., Triggs, B., Jurie, F., 2007. Fast discriminative visual codebooks using randomized clustering forests. In: NIPS.

Murphy, R. F., Velliste, M., Porreca, G., 2003. Robust numerical features for description and classification of subcellular location patterns in fluorescence microscope images. Journal of VLSI signal processing systems for signal, image and video technology 35 (3), 311–321.

Myers, G., 2012. Why bioimage informatics matters. Nature methods 9 (7), 659–660.

Nillius, P., Sullivan, J., Carlsson, S., 2006. Multi-target tracking-linking identities using bayesian network inference. In: CVPR.

Nowozin, S., Lampert, C. H., 2011. Structured learning and prediction in computer vision. Now publishers Inc.

Okuma, K., Taleghani, A., De Freitas, N., Little, J. J., Lowe, D. G., 2004. A boosted particle filter: multitarget detection and tracking. In: ECCV.

Oron, S., Bar-Hillel, A., Levi, D., Avidan, S., 2012. Locally orderless tracking. In: CVPR.

Otsu, N., 1975. A threshold selection method from gray-level histograms. Automatica 11 (285-296), 23–27.

Pele, O., Werman, M., 2009. Fast and robust earth mover's distances. In: CVPR.

Peng, H., 2008. Bioimage informatics: a new area of engineering biology. Bioinformatics 24 (17), 1827–1836.

Peng, H., Ruan, Z., Long, F., Simpson, J. H., Myers, E. W., 2010. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. Nature biotechnology 28 (4), 348–353.

Pepperkok, R., Ellenberg, J., 2006. High-throughput fluorescence microscopy for systems biology. Nature Reviews Molecular Cell Biology 7 (9), 690–696.

Platt, J. C., 1998. Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. rep., Advances In Kernel Methods - Support Vector Leraning.

Rasmussen, C. E., 2006. Gaussian processes for machine learning. The MIT Press.

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

Ren, X., Malik, J., 2007. Tracking as repeated figure/ground segmentation. In: CVPR.

Reymann, J., Beil, N., Beneke, J., Kaletta, P. P., Burkert, K., Erfle, H., Oct 2009. Next-generation 9216-microwell cell arrays for high-content screening microscopy. BioTechniques 47 (4), 877–878.

Rios, J., Ross, K., 2010. Massively parallel dantzig-wolfe decomposition applied to traffic flow scheduling. Journal of Aerospace Computing, Information, and Communication 7 (1), 32–45.

Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.-Y., 2011. Density-aware person detection and tracking in crowds. In: ICCV.

Roller, B., 2004. Max-margin markov networks. In: NIPS.

Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review 65 (6), 386.

Roth, D., Yih, W., 2007. Global inference for entity and relation identification via a linear programming formulation. In: Introduction to Statistical Relational Learning. The MIT Press, pp. 553–580.

Rother, C., Kolmogorov, V., Lempitsky, V., Szummer, M., 2007. Optimizing binary mrfs via extended roof duality. In: CVPR.

Rubner, Y., Tomasi, C., Guibas, L. J., 1998. A metric for distributions with applications to image databases. In: ICCV.

Rubner, Y., Tomasi, C., Guibas, L. J., 2000. The earth mover's distance as a metric for image retrieval. In: ICCV.

Russell, B. C., Torralba, A., Murphy, K. P., Freeman, W. T., 2008. Labelme: a database and web-based tool for image annotation. International journal of computer vision 77 (1-3), 157–173.

Ryan, D., Denman, S., Fookes, C., Sridharan, S., 2009. Crowd counting using multiple local features. In: Digital Image Computing Techniques and Applications.

Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H., 2009. On-line random forests. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE, pp. 1393–1400.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al., 2012. Fiji: an open-source platform for biological-image analysis. Nature methods 9 (7), 676–682.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., Williamson, R. C., 2001. Estimating the support of a high-dimensional distribution. Neural computation 13 (7), 1443–1471.

Schölkopf, B., Smola, A. J., 2002. Learning with kernels. The MIT Press.

Schrijver, A., 2003. Combinatorial optimization: polyhedra and efficiency. New York: Springer-Verlag.

Schulter, S., Wohlhart, P., Leistner, C., Saffari, A., Roth, P. M., Bischof, H., 2013. Alternating decision forests. In: CVPR.

Sharp, T., 2008. Implementing decision trees and forests on a gpu. In: ECCV.

Shi, L., Reid, L. H., Jones, W. D., Shippy, R., Warrington, J. A., Baker, S. C., Collins, P. J., De Longueville, F., Kawasaki, E. S., Lee, K. Y., et al., 2006. The microarray quality control (maqc) project shows inter-and intraplatform reproducibility of gene expression measurements. Nature biotechnology 24 (9), 1151–1161.

Shitrit, B., Berclaz, J., Fleuret, F., Fua, P., 2013. Multi-commodity network flow for tracking multiple people. TPAMI.

Shotton, J., Johnson, M., Cipolla, R., 2008. Semantic texton forests for image categorization and segmentation. In: CVPR.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R., 2013. Real-time human pose recognition in parts from single depth images. Communications of the ACM 56 (1), 116–124.

Simon, J. C., Dickinson, M. H., 2010. A new chamber for studying the behavior of drosophila. PLoS One 5 (1), e8793.

Small, K., Roth, D., 2010. Margin-based active learning for structured predictions. International Journal of Machine Learning and Cybernetics 1 (1-4), 3–25.

Smola, A. J., Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and computing 14 (3), 199–222.

Smolensky, P., 1986. Information processing in dynamical systems: Foundations of harmony theory. Department of Computer Science, University of Colorado, Boulder.

Sokolowski, M. B., 2001. Drosophila: genetics meets behaviour. Nature Reviews Genetics 2 (11), 879–890.

Sommer, C., Fiaschi, L., Hamprecht, F., Gerlich, D., 2012. Learning-based mitotic cell detection in histopathological images. In: ICPR.

Sommer, C., Straehle, C., Kothe, U., Hamprecht, F., 2011. Ilastik interactive learning and segmentation toolkit. In: ISBI.

Sra, S., Nowozin, S., Wright, S. J., 2011. Optimization for Machine Learning. The MIT Press.

Sumpter, D. J., 2006. The principles of collective animal behaviour. Philosophical Transactions of the Royal Society B: Biological Sciences 361 (1465), 5–22.

Swedlow, J. R., Goldberg, I. G., Eliceiri, K. W., et al., 2009. Bioimage informatics for experimental biology. Annual review of biophysics 38, 327.

Tang, S., Andriluka, M., Schiele, B., 2012. Detection and tracking of occluded people. In: BMVC.

Tao, P. D., An, L. T. H., 1997. Convex analysis approach to dc programming: theory, algorithms and applications. Acta Mathematica Vietnamica 22 (1), 289–355.

Tax, D. M., Duin, R. P., 1999. Data domain description using support vectors. In: ESANN. Vol. 99. pp. 251–256.

Teo, C. H., Vishwanthan, S., Smola, A. J., Le, Q. V., 2010. Bundle methods for regularized risk minimization. The Journal of Machine Learning Research 11, 311–365.

Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., 2006. Large margin methods for structured and interdependent output variables. Journal of Machine Learning Research 6 (2), 1453.

Vanderbei, R. J., 2008. Linear programming. Springer.

Viola, P., Jones, M. J., 2004. Robust real-time face detection. International journal of computer vision 57 (2), 137–154.

Wählby, C., Riklin-Raviv, T., Ljosa, V., Conery, A. L., Golland, P., Ausubel, F. M., Carpenter, A. E., 2010. Resolving clustered worms via probabilistic shape models. In: ISBI.

Wainwright, M. J., Jaakkola, T. S., Willsky, A. S., 2005. Map estimation via agreement on trees: message-passing and linear programming. Information Theory, IEEE Transactions on 51 (11), 3697–3717.

Wang, F., Guibas, L. J., 2012. Supervised earth mover's distance learning and its computer vision applications. In: CVPR.

Yu, C.-N. J., Joachims, T., 2009. Learning structural svms with latent variables. In: ICML.

Yuille, A. L., Rangarajan, A., 2003. The concave-convex procedure. Neural Computation 15 (4), 915–936.

Zhang, L., Li, Y., Nevatia, R., 2008. Global data association for multi-object tracking using network flows. In: CVPR.