Reconstruction of Microtubule Centerlines from Electron Tomograms

Dissertation zur Erlangung des Grades
eines Doktors der Naturwissenschaften (Dr. rer. nat.)
am Fachbereich Mathematik und Informatik
der Freien Universität Berlin

vorgelegt von

Britta Weber

Berlin, 2013

Erstgutachter:	Prof. Dr. Knut Reinert Freie Universität Berlin Institut für Informatik Algorithmische Bioinformatik
Zweitgutachter:	Dr. Eugene Myers Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG)

Tag der Disputation: 15.7.2015

Eidesstattliche Erklärung

Gemäß §7 (4) der Promotionsordnung versichere ich hiermit, diese Arbeit selbstständig verfasst zu haben. Ich habe alle bei der Erstellung dieser Arbeit benutzten Hilfsmittel und Hilfen angegeben.

Berlin, den 7.10.2013

Britta Weber

Synopsis

Understanding the mechanisms of cell division is an active research topic in molecular cell biology. Of particular interest is the organization of the mitotic spindle, a structure that separates the chromosomes during cell division. The mitotic spindle is composed of microtubules, elongated tubular macromolecules with a diameter of $\sim 25\,\mathrm{nm}$. The only imaging technique that is available to a wide community and provides the required resolution to capture details about microtubules is electron microscopy. Electron microscopy can be extended to allow volumetric imaging of samples, in which case it is called electron tomography. However, the automatic detection of microtubules in electron tomograms is a difficult task due to the low contrast of the data. Image processing methods for an automatic analysis of the tomograms are not available. Furthermore, the thickness of samples that can be imaged is limited to 300 nm. Thicker samples have to be cut into thin sections before electron tomography can be applied. Software for automatically stitching these sections has not been developed either.

Because automated tools for segmentation and stitching of microtubules are not available, both tasks have to be performed manually. Unfortunately, manual segmentation is time consuming for large samples with a dense microtubule network. In addition, manual stitching of the tomograms is often infeasible because the image data does not always contain enough concise objects that would allow an expert to find correspondences. Therefore, conclusions drawn from electron tomographic data is currently mostly based on either small samples containing few microtubules or single sections of complex structures. Consequently, although the mitotic spindle has been subject to intense research for over a century, simple properties, such as the length of microtubules in the spindle or their number, are still unknown for most model organisms.

In this thesis, we present methods for 1) an automatic segmentation of microtubule centerlines in electron tomograms, and 2) an automatic stitching of the lines in serial sections. Because of the difficulty of these tasks, we focus on methods that are flexible enough to allow the integration of detailed expert knowledge. For the centerline segmentation, we use 3D template matching, and exploit both knowledge about shape of microtubules and microscopy artifacts to design the templates. For the stitching of the segmented lines, we make use of statistical methods, because not only are these methods designed to facilitate the inclusion of prior knowledge but they also naturally deal with noise and outliers. For the registration of the lines, we present a way to model the orientation of lines as a mixture of Fisher-Mises distributions. With this formulation, lines can be registered with an unsupervised learning technique that estimates transformation parameters with the expectation maximization algorithm. Furthermore, we present a semi-automatic approach to find the final connections of the lines in two consecutive sections. The line matching problem is formulated in terms of a probabilistic graphical model. To find the correct correspondences of line ends, we use belief propagation. We handle the poor convergence properties of this algorithm by detecting ambiguous and conflicting assignments of lines automatically. An expert can then influence the final output of the algorithm by solving conflicts manually.

A detailed error analysis on true biological data and assessment of the reliability of the results is the prerequisite for analyzing the resulting line representations of the microtubules. To this end, the developed workflow for segmenting and stitching of microtubule centerlines is evaluated on plastic-embedded samples of *C. elegans* early embryos and of spindles from *X. laevis* egg extracts. Our results suggest that the output of the presented algorithms together with little manual correction is of sufficient quality to allow a detailed analysis of dense microtubule networks. Finally, we exemplarily show results for the centrosome of a *C. elegans* mitotic spindle.

Zusammenfassung

Die Untersuchung von Zellteilungsmechanismen ist ein wichtiger Schwerpunkt der molekularen Zellbiologie. Von besonderem Interesse ist der Aufbau der mitotischen Spindel, einer Struktur die maßgeblich an der Separation der Chromosomen während der Zellteilung beteiligt ist. Die mitotische Spindel besteht aus röhrenförmigen Makromolekülen, genannt Mikrotubuli, die einen Durchmesser von nur $\sim 25\,\mathrm{nm}$ haben. Das einzige bildgebende Verfahren, das einer großen Gemeinschaft zugänglich ist und die nötige Auflösung besitzt, um Details der Mikrotubuli zu erfassen, ist Elektronenmikroskopie. Elektronenmikroskopie kann auch genutzt werden, um volumetrische Aufnahmen von Proben zu gewinnen. Diese Technik heißt Elektronentomographie. Mikrotubuli in solchen Volumen automatisch zu detektieren, ist aufgrund des schlechten Kontrastes der Daten schwierig. Bildverarbeitungsverfahren für eine automatische Analyse der Tomogramme sind derzeit noch nicht vorhanden.

Weiterhin darf die aufzunehmende Probe nicht dicker als 300 nm sein. Dickere Proben müssen in dünne Schichten zerschnitten werden, bevor man sie mit Elektronentomographie aufnehmen kann. Software, um diese Schnitte automatisch wieder zusammenzusetzen, existiert ebenfalls nicht.

Da es für beide Probleme, Segmentierung und Zusammensetzen der Daten, keine automatisierte Lösung gibt, wird beides bis zum heutigen Zeitpunkt manuell gelöst. Für große Datensätze mit einem dichten Mikrotubulinetzwerk ist die manuelle Segmentierung von Elektronentomogrammen sehr zeitaufwändig. Zusätzlich können Tomogramme nicht immer manuell zusammengesetzt werden, da die Bilddaten oft nicht genug herausragende Strukturen beinhalten, um Korrespondenzen zwischen zwei Schichten zu finden. Aus diesen Gründen können Erkenntnisse mittels Elektronentomographie momentan nur aus kleinen Proben, die wenige Mikrotubuli enthalten, oder aus einzelnen Schichten von komplexeren Strukturen gewonnen werden. Obwohl die mitotische Spindel nun schon seit über hundert Jahren Gegenstand der Forschung ist, sind einfache Eigenschaften wie Länge der Mikrotubuli in der Spindel oder ihre Anzahl noch immer nicht bekannt.

In dieser Arbeit präsentieren wir Methoden zur 1) automatischen Segmentierung der Mittellinien von Mikrotubuli und 2) zum automatischen Zusammensetzen der Linien in aufeinanderfolgenden Schichten. Schwerpunkt ist dabei die Verwendung von Verfahren, die flexibel genug sind, um Berücksichtigung von detailiertem Expertenwissen zuzulassen. Um die Mittellinien zu segmentieren, verwenden wir 3D-Template-Matching und nutzen dabei sowohl Wissen über die Form von Mikrotubuli als auch Wissen über Mikroskopieartefakte, um das Template zu erzeugen. Um die segmentierten Linien zusammenzusetzen, nutzen wir vor allem statistische Methoden, da diese nicht nur natürlich die Integration von Expertenwissen zulassen, sondern zusätzlich auch oft automatisch mit Rauschen und Ausreißern umgehen. Um die Linien zunächst zu registrieren, modellieren wir die Orientierung der Linien als Fisher-Mises-Mischverteilung. Mit dieser Formulierung können die Linien mittels unüberwachten Lernens registriert werden. Wir nutzen den Expectation-Maximization-Algorithmus um Transformationsparameter abzuschätzen. Weiterhin beschreiben wir einen semiautomatischen Ansatz, um Linien in zwei aufeinanderfolgenden Schichten zu verbinden. Wir formulieren das Problem als Probabilistisches Graphisches Modell. Um die richtigen Zuordnungen zu finden, nutzen wir den Belief-Propagation-Algorithmus. Um Konvergenz dieses Algorithmus zu erzwingen, detektieren wir Konflikte in Zuweisungen automatisch. Ein Experte kann Konflikte so manuell auflösen und damit auch die Qualität des Endergebnisses positiv beeinflussen.

Eine detaillierte Fehleranalyse auf biologischen Datensätzen ist die Voraussetzung, um die Linienrepräsentation der Mikrotubuli weiter zu analysieren. Zu diesem Zweck werden die entwickelten Verfahren zum Segmentieren und Zusammensetzen auf Proben von *C. elegans* Embryos und *X. laevis* Eiextrakt evaluiert. Unsere Resultate weisen darauf hin, dass die entwickelten Verfahren zusammen mit wenig manueller Interaktion von ausreichender Güte sind, um eine detaillierte Analyse dichter Mikrotubulinetzwerke zu erlauben. Schließlich zeigen wir noch beispielhaft Ergebnisse für ein Zentrosom einer mitotischen Spindel in einem *C. elegans* Embryo.

Contents

Chapter 1

Introduction

Imaging subcellular processes with electron microscopy has become a standard technique in biological research during the last century. However, analyzing images obtained by electron microscopy remains a time-intensive procedure due to the lack of dedicated software for automated segmentation of biological structures in electron microscopic images. In this thesis, we are mainly concerned with the development of algorithms for an automatic reconstruction of microtubule centerlines in electron tomograms. This chapter gives a basic overview on the background and the objectives of this thesis.

1.1 Background

Microtubules, together with actin filaments and intermediate filaments, form the cytoskeleton of eukaryotic cells. Here, we review only the major facts about these protein structures as described by Alberts et al. [1] and refer the reader to the same book and references therein for details on the matter. A microtubule is a macromolecule, which is composed of tubulin. This protein polymerizes to form a tubular shape with a diameter of approximately 25 nm. The length of these tubes ranges from a few nanometers to several micrometers. Microtubules serve many purposes: For example, they stabilize the cells shape and are used by other organelles as transport routes. In this thesis, however, the main focus lies on their role in cell division. When an eukaryotic cell divides, microtubules form an assembly called the mitotic *spindle*. The spindle is a bipolar network of microtubules that attaches to chromosomes during cell division. Protein motors then attach to the microtubules and cause anti-parallel sliding which ultimately pulls the chromosomes apart (see, for example, Mogilner and Craig [2]). The rough shape of the spindle in different stages of mitotic cell division and its function is illustrated in Figure 1.1. The complex mechanisms that steer microtubule polymerization and organization are subject to intense research.

Due to their small diameter (25 nm), imaging microtubules in cells is a challenging task. The achievable resolution with conventional light microscopy techniques is > 100 nm and therefore insufficient to resolve individual microtubules. Currently, new developments in optical theory and imaging allow breaking this resolution limit (see Section 2.2.1 for more details on light microscopy imaging). While these techniques are on the verge of providing high resolution imaging of selected structures in cells on a nanometer scale, they are not yet capable of imaging dense microtubule assemblies such as the spindle in a quality sufficient for an analysis of the exact organization of the thin filaments. Light microscopy is, for example, used to quantify the amount of polymerized tubulin in a cell, to get an overview of the structure of assembled microtubules or to observe individual microtubules in sparse networks. However, to fully understand the mechanisms of spindle organization, detailed information on, for example, length, number and mutual distance of the microtubules is required.

The major imaging technique for acquiring high resolution images of microtubule assemblies is electron microscopy. In this thesis, we will only consider a single variant of electron microscopy, called transmission electron microscopy. Here, similar to conventional light microscopy, samples are rayed with an electron beam and the resulting image shows a projection of the sample. Transmission electron microscopy can also be used to obtain 3D volumes of a sample. To achieve this, the

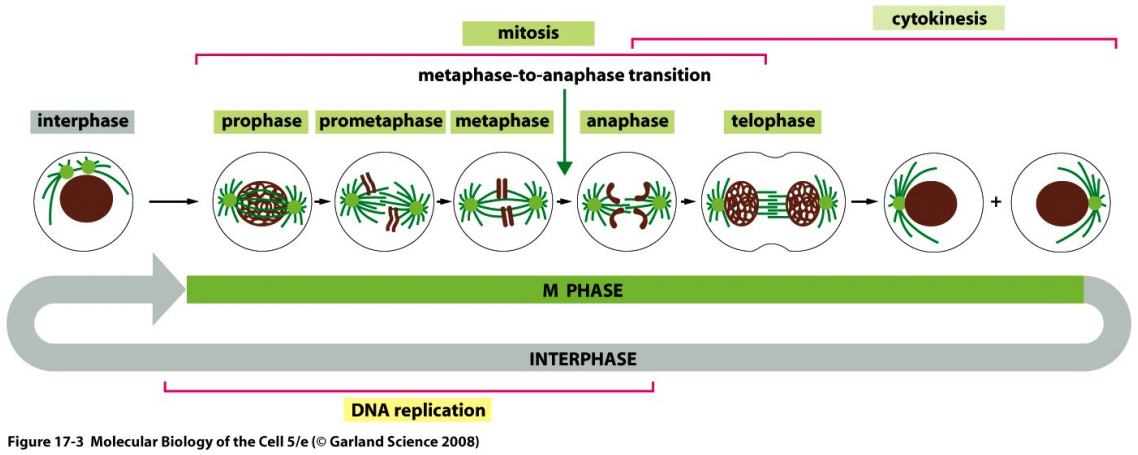


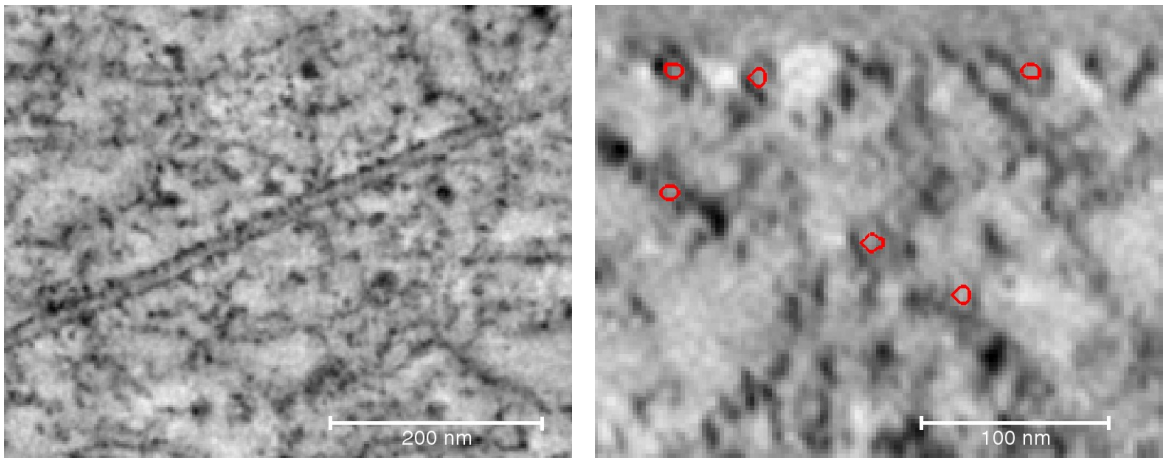
Figure 1.1. Illustration of the mitotic cell division cycle. Nucleus and chromosomes are depicted in brown. The light green circles indicate the position of the centrosomes, which are assumed to be the microtubule organizing center. Microtubules are indicated by dark green lines. Here, only the general shape of the spindle is sketched. Depending on species and cell, spindles can contain thousands of microtubules. Reproduced from Alberts et al. [1] with kind permission of Garland Science, Taylor & Francis Group.

sample is tilted under the beam and projections at different tilt angles are acquired. A volumetric reconstruction of the sample, called an *electron tomogram*, is then computed from these projections. This technique is called *electron tomography*. Electron tomography is currently the only imaging method that allows a full 3D reconstruction of a sample with the resolution required for gaining detailed insight into dense microtubule networks such as the mitotic spindle. Electron tomograms typically show a resolution of approximately 5 nm (see Section 2.2.2 for a detailed description of the technique).

However, imaging samples with electron tomography is also a difficult task and has two major disadvantages: 1) Samples must be cut in sections of about 300 nm thickness since thicker sections are hard to image with this technique (see Egerton [3] and Section 2.2.2). To obtain a full 3D volume of thick samples, individual tomograms of physical sections must be prepared, which later have to be combined in order to obtain a complete volume. 2) The volumes are typically of a low quality: The thin sections cannot be tilted up to 90° when acquiring the projections. About 30% of the projections are missing. Therefore, the reconstructed tomograms show an artifact called the *missing wedge*, which causes an anisotropic resolution in the volumes (see Section 2.2.2) and the appearance of objects in the resulting volumes is distorted. In addition, contrast, resolution and signal-to-noise ratio are influenced by the outcome of several steps in the highly sophisticated sample preparation protocol (see Section 2.2.3). Figure 1.2 shows a typical electron tomogram of microtubules and illustrates the impact of the missing wedge artifact.

These effects render the analysis of microtubule network properties in electron tomograms, such as length of the filaments and their spatial distribution, difficult. For such an analysis, two steps have to be performed: 1) The microtubules have to be segmented, that is, a line representation of their shape must be obtained, and 2) serial sections of a sample have to be combined to one dataset. These steps are illustrated in Figure 1.3.

To date, the segmentation of microtubules has been a manual process due to the lack of designated image processing software. Unfortunately, manual segmentation of microtubules is very time consuming and currently the major limitation when aiming at a statistical analysis of large microtubule networks. For example, the midsection of a centrosome in the mitotic spindle of an early *C. elegans* embryo contains between 1500 and 3000 microtubules. With current methods, manual segmentation of centerlines in one of these sections can take several hours (see Weber et al. [4]) depending on the quality of the tomogram. At least 20 of these sections are needed to cover a full centrosome in such a sample. Consequently, a biologist would have to spend roughly



(a) (b)

Figure 1.2. (a) A transversal slice of an electron tomogram. Microtubules are visible in such a slice, but they are surrounded by clutter and the cross section through the tubular shape here does not appear as clear parallel lines. (b) A coronal slice through the same tomogram. Microtubules (marked in red) hardly appear as clear rings as would be expected for their tubular shape. The missing wedge artifact causes distortions and a decrease in resolution of lateral and sagital slices to nearly half the resolution of transversal slices.

one person month only on the segmentation to get a reconstruction of all microtubules in a single centrosome. This is the reason why in studies that describe microtubule properties using electron tomography, results are mostly based on single observations in individual samples as opposed to a statistical analysis of many samples (see, for example, Hoog et al. [5]), Srayko et al. [6], O'Toole et al. [7]).

Software for the automatic stitching of electron tomograms is also not readily available. Semi-automatic approaches have been developed, but these require that experts identify matching regions in the tomograms and place landmarks that can be used for registration of the volumes. Depending on the size of the tomogram, many of such landmarks have to be placed, because preparation and imaging of a sample causes anisotropic deformations of the material which must be compensated for. Each of these landmarks has to be precisely positioned because in order to connect microtubules of different sections, their ends must not be further apart than a few nanometers. This is a cumbersome and often infeasible task. Therefore, electron tomograms are rarely stitched.

For these reasons, detailed quantitative measurements on, for example, number and length of microtubules in a spindle are still not available, except for model organisms having a small spindle size comparable to that of yeast (see, for example, Ding et al. [8]).

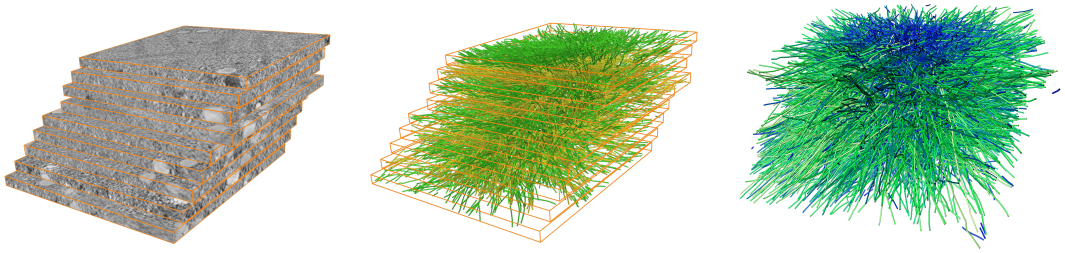
1.2 Objective of this thesis

In this thesis, we develop methods that allow a full reconstruction of the microtubule network from serial electron tomograms of a sample. We present methods for automatic

- Microtubule segmentation, and
- Stitching of consecutive sections.

The goal of these two steps is to obtain from a stack of electron tomograms a set of polygonal lines each of which represents a microtubule centerline that is present in the tomograms (see Figure 1.3(c)).

The major challenge for an automization is the required accuracy of the result: For example, the number of microtubules can only be estimated if the computed line representation contains few breaks in the reconstructed lines. Each discontinuity of a line would cause a microtubule



(a) (b) (c)

Figure 1.3. (a) Electron tomography of a sample requires cutting the sample into thin sections. For each of these sections a tomogram is prepared. (b) To get a geometric representation of the microtubules, the first task is a segmentation of their centerlines. The result for each individual tomogram is a set of polygonal lines. Each of the lines represents the centerline on a microtubule. We use these lines to stitch the data and obtain a full 3D reconstruction of the microtubule network. (c) The aligned and stitched line sets. The color indicates the length of the computed microtubule centerlines. Long lines are colored in green and short lines are colored in blue.

to be counted twice and furthermore disturb any measurement of length of microtubules. These requirements are in strong contrast to the quality of the input data: Microtubules seldom appear as black tubes on white background. Instead, they are subject to artifacts such as intensity variations and the missing wedge. Furthermore, they are embedded in the cytoplasm, which appears as strong clutter around the tubes. This makes it hard to segment microtubules reliably at their full length. The stitching is also challenging: Microtubules often have a small mutual distance of less than 40 nm which can make decisions on the continuation of a line from one section to the next difficult. This problem is further complicated by the fact that preparing and imaging causes anisotropic deformation of the sample.

Our objective is therefore to develop algorithms for both performing the segmentation of microtubules and the stitching of consecutive sections automatically and tools for visual inspection and manual post-processing of the results. Furthermore, we provide a detailed error analysis and an assessment of the reliability of the results on true biological data. This is essential if conclusions are to be drawn from the obtained data. To measure the performance of the proposed methods, the developed workflow for segmenting and stitching will be evaluated on plastic-embedded samples of several model organisms and different sample preparation and image acquisition modalities. The ultimate goal in this thesis is to build an application, that can be used for the analysis of electron tomograms in biological research.

1.3 3D reconstructions of microtubule networks in biology

The spindle has been studied for more than 150 years but despite recent advances in microscopy and experimental biology, simple questions about its organization remain unanswered. The most widely accepted model for microtubule organization in the spindle is depicted in Figure 1.4(a). This model describes three categories of microtubules: Astral microtubules that emerge from the centrosome and orient the spindle, kinetochore microtubules connecting centrosome and chromosomes and interpolar microtubules which do not attach to the chromosomes. In this model, interpolar and kinetochore microtubules are responsible for the movement of the chromosomes. Protein motors such as dyenin and kinesin connect to two microtubules and slide them in an anti-parallel fashion to generate the pulling force (Figure 1.4(b)). While observations from electron and light microscopy support this model, the detailed structure of the spindle has never been fully observed in an image.

Because no precise measurements can be taken from light microscopy imaging and electron tomography is too expensive without further automation of the analysis, this model remains theoretical in nature and is subject to controversy (Mogilner et al. [2], Needleman and Farhadifar [9]). For example, it is yet unclear if microtubules even stretch the whole way from centrosome to chro-

mosomes. Instead, in some species cross linking short microtubules might stabilize the spindle (Yang et al. [10]). This model is depicted in Figure 1.4(c). The organization might even vary from species to species (Karsenti and Vernos [11]). A means to obtain a detailed geometrical representation of all microtubules in the spindle from true biological data would allow to immediately assess different theories for their validity.

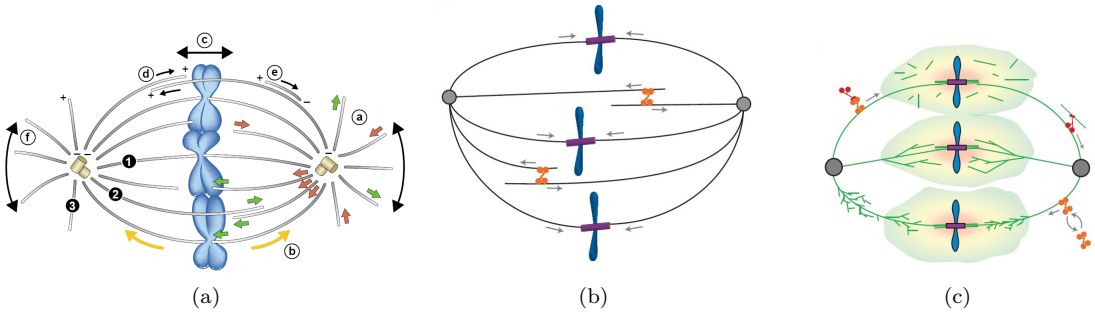


Figure 1.4. (a) Function of microtubules in the spindle as described in Wittmann et al. [12]. Microtubules can be divided roughly into three categories: kinetochore microtubules (1), interpolar microtubules (2), and astral microtubules (3). (a): microtubules undergoing dynamic instability, where green arrows indicate growing and red arrows shrinking of the microtubules. The spindle tears the chromosome sets apart ((b) and (c)). Furthermore, microtubules movement is probably triggered by anti-parallel sliding induced by protein motors (d) and microtubules serve as means of dyenin-dependent transport (e). Astral microtubules orient the spindle (f). Reproduced from Wittmann et al. [12] with kind permission of the Nature Publishing Group. (b) Classical view of the function of kinetochore and interpolar microtubules. Kinetochore microtubules attach to the chromosomes via kinetochores. Interpolar microtubules reach beyond the chromosomes. Microtubules are linked via protein motors that induce an anti-parallel sliding and thereby cause the chromosomes to move. (c) A competing model of spindle organization. Here, microtubules do not span from MTOC to chromosome but instead form a dense net of shorter tubes that are cross linked to maintain stability of the spindle. Figures reproduced from Mogilner and Craig [2] with kind permission of the Journal of Cell Science.

The ability to quickly gather exact microtubule measurements from samples would also allow for a statistical analysis of the influence of certain proteins on spindle building for example, in studies comparing wild type and mutant samples (gene inactivation by RNA interference, RNAi). For example, Greenan et al. used an RNAi approach to show that centrosome size determines the length of the mitotic spindle in *C. elegans* (Greenan et al. [13]). However, detailed information, such as the influence of the experiment on length and the number of microtubules at the centrosome has not been studied directly.

Furthermore, quantitative measurements are essential for predicting microtubule behavior with computer simulations. Parameters like length, number and mutual distance can serve as input for biophysical simulations that make assumptions on these properties (see, for example, Paul et al. [14], Loughlin et al. [15] and Mogilner et al. [16] for a review on spindle assembly simulations). Finally, a geometric representation extracted from biological data is needed as a baseline for assessing the validity of mathematical models and computer simulations of spindle assembly (Mogilner et al. [16]).

1.4 Outline of this thesis

This thesis consists of five chapters, including this introduction.

Chapter 2 , *Biological and technical background*, explains details on the biological background of this work, which was only briefly sketched in this introduction. We start by describing the basic structure and function of microtubules in cells. Then, different imaging techniques

used for studying microtubules are discussed. In particular, we explain in detail the basics of electron tomography and give an overview over image modalities and artifacts we observed in biological data. We then describe origin and preparation details of the samples that we used for evaluation and furthermore introduce the model organisms used throughout this thesis. The chapter closes with a discussion of the problems arising for both biologists working with electron tomography and computer scientists aiming at an automatic segmentation of microtubules and stitching of serial sections.

Chapter 3 , *Segmentation of microtubule centerlines in electron tomograms*, describes a semi-automatic approach to centerline segmentation. We begin with reporting recent approaches to tackle this problem.

In the following two sections, we explain details of the method that we used for microtubule centerline extraction. The method has two steps: 1) Microtubules are enhanced in the volumes using 3D template matching, and 2) centerlines are traced with a simple heuristic in the enhanced volumes. Furthermore, we describe a user interface to facilitate a quality control and subsequent manual corrections.

The main part of this chapter is dedicated to the evaluation of the proposed method by comparing results of the automatic approach to manually segmented lines. We describe our quality measure, how we generate a ground truth from several manually segmented lines on the same volume, and how we assess the sensitivity of the parameters of the automatic tracing method. We then evaluate the performance of the tracing algorithm on more than 50 datasets of *C. elegans* early embryos. Finally, we discuss the usefulness of the proposed method for application in biological research.

Chapter 4 , *Stitching tomogram sections*, describes how we stitch the line sets computed as described in the previous chapter to obtain a full 3D reconstruction of the microtubule centerlines in a sample. Our approach consists of three steps: 1) We find an initial transformation using a distance compatibility graph. 2) We register the line ends using the coherent point drift algorithm (Myronenko et al. [17]), which we extend to handle orientation of lines by introducing a periodic random variable to the probabilistic formulation. 3) We compute the final line end correspondences by modeling the problem in terms of a Markov network and compute the best matching of line ends with belief propagation on a factor graph. Because belief propagation is not guaranteed to converge to a minimum, we show how to identify critical regions and enforce convergence with little user input. We evaluate the performance of the presented methods again by comparing the stitching computed by the proposed algorithms to manually obtained results on biological data. We conclude this chapter with a discussion on the applicability of the presented methods in biological research.

In Chapter 5 , *Microtubule arrangement in the centrosome*, we show how the computed line sets can be analyzed to answer biological questions concerning microtubule arrangement. We apply the whole segmentation and stitching pipeline to tomograms of sections taken from a *C. elegans* early embryo during mitosis. From the resulting 3D line representation, we derive preliminary measurements of centrosome size and the number of microtubules and furthermore describe structural properties of the centrosome.

Chapter 6 , *Conclusion*, summarizes our contributions and findings and closes with a critical discussion on the value of this work for biological research.

Chapter 2

Biological and technical background

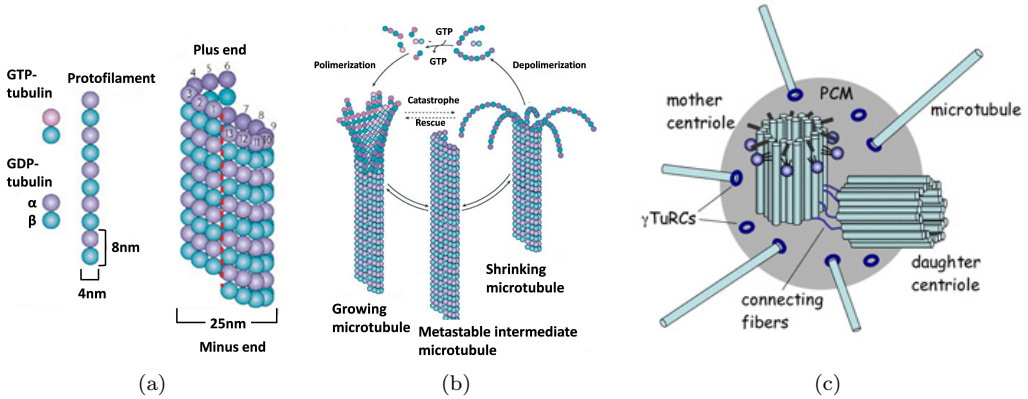
This chapter will introduce the reader to the biological context of this thesis. We start by describing the basics of microtubule assembly, cell division and the mitotic spindle in Section 2.1. In Section 2.2, we introduce different microscopy techniques that are used to study cell division. After reviewing the history of observing microtubules with light microscopy techniques in Section 2.2.1 we explain the basic concept of electron microscopy and electron tomography in Section 2.2.2. Because electron tomographic volumes are the input for our analysis pipeline, we describe in detail properties and defects of electron tomograms in Section 2.2.3. In Section 2.3, the model organisms used throughout this thesis are introduced. This chapter closes with a summary of the problems that are encountered when attempting to analyze microtubule organization using electron tomography (Section 2.4).

2.1 Microtubule assembly during cell division

The cytoskeleton is an assembly of filamentous structures in cells that scaffold the cell and are involved in many cellular processes, such as cell movement, transport of other organelles or cell division in eukaryotic cells. The cytoskeleton consists of three types of filaments: actin, intermediate filaments and microtubules. The basic structure of a microtubule is depicted in Figure 2.1(a). In this thesis, we will be exclusively concerned with microtubules. Microtubules are tubular shaped assemblies of proteins called α and β tubulin (purple and blue in Figure 2.1(a)). These two types of tubulin dock to form heterodimers which in turn form strings that are called protofilaments (Howard and Hyman [18]). Usually, 13 of these protofilaments are arranged to form a tube with a diameter of approximately 25 nm. This tubular structure is called a microtubule. The length of a microtubule can vary from a few nanometers to several micrometers. The microtubule is always constructed from the same heterodimers and one end exposes α-tubulin and the other β-tubulin. The former end is called the minus and the latter the plus end (Alberts et al. [1]).

Both tubulin types have a binding site for Guanosine triphosphate (GTP). β-tubulin can also bind to Guanosine diphosphate (GDP) instead. Depending on which molecule β-tubulin is bound to, the plus end of a microtubule is stable and can grow, or it becomes unstable and depolymerizes as depicted in Figure 2.1(a). The transition from stable to unstable state is called catastrophe and the transition from unstable to stable state rescue (see, for example, Akhmanova and Steinmetz [19]). Microtubules can also grow and shrink slowly from the minus end, but polymerization and depolymerization at the plus end is much faster (Alberts et al. [1]). The exact mechanism of polymerization and depolymerization is currently subject to intense research and different models compete (see Margolin al. [20], Kueh and Mitchison [21]).

Microtubule have many roles in cells. For example, they form the backbone of the flagellum or cilium of single cell organisms like *Trypanosoma brucei* and *Clamydomonas reinhardii* (Alberts et al. [1]). Here, microtubules polymerize once in several doublets or triplets and then remain stable. Microtubules can form skeletons for cells, for example, the sub-pellicular microtubule skeleton of



(a) (b) (c)

Figure 2.1. (a) Schematic of the structure of microtubules. α and β tubulin assemble to dimers. These dimers form protofilaments which in turn form a tube. The end exposing β tubulin is the plus end and the other end exposing α tubulin the minus end. (b) A sketch of microtubule polymerization and depolymerization. Depending on if GTP or GDP is bound to the β-tubulin at the plus end, the microtubule is stable and can grow (left side) or shrink (right side). Figures (a) and (b) are adapted from Akhmanova and Steinmetz [19] with kind permission of the Nature Publishing Group. (a) The basic organization of the centrosome. Centrioles are surrounded by the pericentriolar material which contains the γ-tubulin binding sites for microtubules. Image courtesy of Jens Lüders, IRB Barcelona.

Trypanosoma brucei (Farr and Gull [22], Gull [23]). Microtubules are also transport routes in cells. For example, kinesin and dynein are protein motors that attach to vesicles and move them along a microtubule through the cell (Alberts et al. [1]). Furthermore, for cell division, a large number of microtubules is assembled in a structure called the mitotic spindle which is believed to pull chromosomes apart (Alberts et al. [1], see also Figure 1.1). We have already discussed the basic geometric outline of the spindle in the introduction. Here, we describe further essential properties.

The spindle is a bipolar structure. The two poles of the spindle are called the centrosomes. A centrosome is an organelle whose protein composition or even shape is not fully known until now (see Nigg and Stearns [24] for a review). The position of the centrosomes throughout the cell cycle are depicted in Figure 1.1 as light green circles. At the heart of the centrosome lie the centrioles. These organelles consist of short microtubules with fixed length that are arranged to form a cylindrical shape. They are the center of most centrosomes in eukaryotic cells. The basic organization of the centrosome is depicted in Figure 2.1(c). Centrioles are surrounded by the pericentriolar material which contains, amongst many other proteins, γ-tubulin which serves as a nucleation site for microtubules at the centrosome (Alberts et al. [1]) . The centrosome is believed to be mainly responsible for the organization of microtubule assembly in the spindle and is for this reason referred to as the microtubule organizing center (MTOC) (Alberts et el. [1]).

The formation and structural reorganization of the spindle is highly dynamic. For example, cell division in *C. elegans* takes only minutes (Bao et al. [25]). In this time, thousands of microtubules (the exact number is still unknown and probably varies from species to species and also the cell development) have to polymerize, undergo the reorganizations depicted in Figure 1.1 and depolymerize again (Woodruff et al. [26]). One particularly interesting property of microtubule organization is that some microtubules in the spindle grow, while others shrink at the same time, as illustrated in Figure 1.4(a). This curious effect is called the microtubule dynamics instability (Burbank and Mitchison [27], Mitchison and Kirschner [28]). A variety of proteins are known to regulate microtubule polymerization. These microtubule associated proteins (MAPs) can bind to the ends, usually the plus ends, the whole filament and either encourage growth or cause catastrophe (see Heald and Nogales [29], Howard and Hyman [18], Gouveia and Akhmanova [30]). To study the effects of selected proteins, microtubules can be artificially grown in a petri dish (Gell et al. [31]). However, not all proteins influencing microtubule polymerization in live organisms are known and in particular their interaction is not fully understood. This is why (Gardner et al. [32])

MAPs are often studied in live cells (Srayko et al. [33], Howard and Hyman [18], Müller-Reichert et al. [34], Karsenti and Vernos [11]).

Both research on structural function and dynamic instability of microtubules requires visualizing samples and measuring the effect of experiments. The most important tool in biological research for achieving such a quantification is the microscope. We will next describe different microscopy techniques for imaging the spindle.

2.2 Imaging microtubules

Imaging microtubules in cells dates back to the 1880 s when Walter Flemming discovered the spindle (Paweletz [35]) with a light microscope and described it in his work "*Zellsubstanz, Kern und Zelltheilung.*" in 1882. In this section, we briefly recapitulate major developments of microscopy techniques since then that lead to new insight in the function of microtubules. We will focus on optical light and electron microscopy. Figure 2.2(a) depicts the basic functionality of both transmission electron and classic light microscopy. With both techniques, a beam, either light or electron, penetrates the sample and beam alterations caused by the sample are captured by a camera or the eye. Beam alteration here does not only mean intensity: For example, material in a sample can cause a measurable phase shift of light. While both light and electron microscopy are similar in function, the most fundamental difference is the resolution each of the techniques can achieve. Figure 2.2(b) gives an overview of biological structures that can be observed with each technique. The resolution of light microscopy is limited to approximately 100 nm due to the wavelength of light. Even though this limit is being pushed by several recent developments in optical theory and new microscopy techniques (see next Section), currently, visualization of single microtubules and in particular their detailed morphology still requires an electron microscope which can have a resolution of 0.1 Å.

This section is not a comprehensive introduction to imaging techniques in biology. It should give the reader an understanding on how to place electron tomography in the field of visualization of biological samples. We deliberately omit all other techniques for visualizing samples such as all forms of scanning microscopy, and all forms of protein analysis.

2.2.1 Imaging microtubules with light microscopy

The first one to discover the mitotic spindle was probably Walter Flemming (Paweletz [35]), who observed it in the 1870s in stained and fixed samples with bright field microscopy with a magnification of 18× (Flemming [36]). In the years to follow, many theories on accurate form (see Figure 2.3(a) for an early example), function and behavior of this structure formed (see Wilson [37] for early theories and Bajer and Molè-Bajer [38] for a review of cytology until 1970s). However, until the 1950s, the spindle could only be imaged in fixed samples and therefore all conclusions about cell division had to be drawn based on observations made on chemically altered cells. In the 1950s the development of polarization microscopy (Figure 2.3(b)) allowed for the first time observing microtubules in life and unstained cells (Inoué [39], Inoué [40], Mitchison [41]). However, with these techniques individual microtubules could not be observed yet. For sparse microtubule networks, the drop of cost of video together with the development of differential interference contrast microscopy [42] allowed analyzing the function of individual microtubules as transport route for organelles as depicted in Figure 2.3(d) (Allen et al. [43], Inoué [44], reviewed in Cassimeris et al. [45]).

The development of fluorescence microscopy that started in the 1910s [46] shed further light on the role of microtubules. Here fluorescent molecules that absorb light at one wavelength and emit it at another (Alberts et al. [1]) are used as a stain. Exploiting this effect for imaging is fundamentally different from classical light microscopy: Proteins or other molecules are tagged with a fluorescent molecule, which is stimulated with a certain light frequency. The light emitted from the fluorescent molecule is then detected. This technique allows capturing the position of selected labeled molecules while all other structures in the cell remain invisible. Fluorescence microscopy is now one of the most widely used imaging technique in biology. Several variants exist: Fluorescent molecules can be docked to specific proteins by antibodies raised against the protein of interest (discovered by Coons at al. [47], see Semenova and Rodionov[48] and Alberts

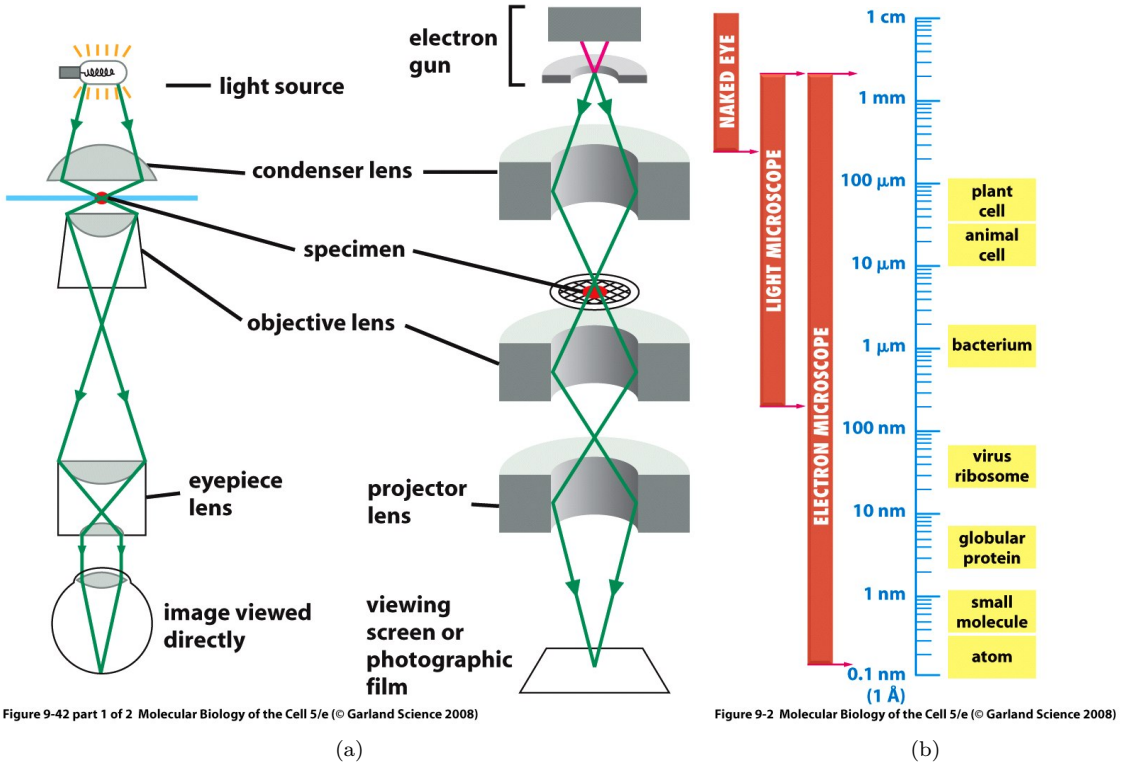


Figure 2.2. (a) General principle of transmission light and electron microscopy. (b) Resolution limits of light and electron microscopy. Both images were reproduced from Alberts et al. [1] with kind permission of Garland Science, Taylor & Francis Group.

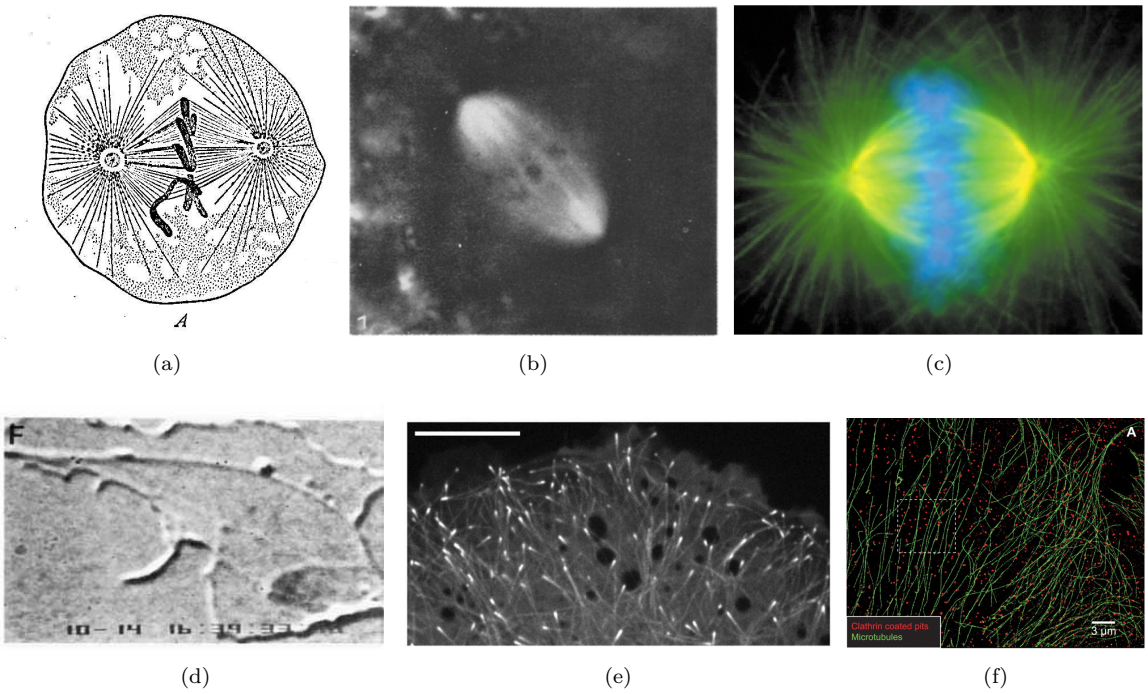
et al. [1] for an overview on the technique). Figure 2.3(c) shows an example of a spindle imaged with immunoflourescence microscopy. Because antibodies bound to fluorescent probes need to be inserted in the cell, observation with this technique is restricted to dead samples. Life samples can be imaged by genetically altering the organism to produce fluorescent molecules (discovered by Taylor and Wang [49], see Goodson et al. [50] for an overview). Using this technique, microtubule dynamics can be measured quantitatively, for example, by measuring fluorescence recovery after photo bleaching (first introduced by Magde et al. [51]), or by observing dynamics over time for example by tracking microtubule ends only as depicted in Figure 2.3(e). Therefore, this technique is well-suited for studying microtubule dynamic instability (Straube [52]).

Accuracy of detailed measurements on dense microtubule networks such as the mitotic spindle is still limited with light microscopy alone, due to the fact that a microtubule's diameter is only 25 nm, which is about five to ten times smaller that the resolution limit of light (Figure 2.2(b)).

However, recently new techniques like structured illumination (Gustaffson [56]), stochastic optical reconstruction microscopy (STORM) (Rust et al. [57]) or photoactivated localization microscopy (PALM)(Beltzig et al. [58]) have broken this resolution limit. Figure 2.3(f) shows an example of microtubules imaged with STORM. Resolution of the new techniques can be down to a few nanometers. See Hell [59] and Fernández-Suárez and Ting [60] for a review on these techniques. These techniques increase resolution but they have not yet succeeded to resolve individual microtubules in dense networks (Loughlin [15]).

2.2.2 Transmission electron microscopy and electron tomography

Transmission electron microscopy (TEM) creates projections of samples by penetrating a sample with an electron beam. Similar to light microscopy, the scattering of electrons and also the phase contrast of the beam after leaving the sample can be measured. For a more thorough introduction to the exact mechanism, see, for example, Williams and Carter [61] and Egerton [3]. In this



(a) (b) (c)

(d) (e) (f)

Figure 2.3. (a) Structure of the spindle according to Theodor Boveri [53] in the 19th century. Reproduced from Wilson [37]. (b) The spindle imaged in a life cell with polarized light microscopy. Reproduced from Inoué [39] with kind permission of Springer.(c) Example of immunofluorescence imaging of the spindle. Reproduced from Wittmann et al. [12] with kind permission of the Nature Publishing Group.(d) A snapshot from a movie of cytoplasmic particles moving along microtubules imaged with video enhanced DIC. Reproduced from Allen et al. [43] with kind permission of John Wiley and Sons.(e) Microtubule tips as observed by tagging the microtubules ends with fluorescent markers. Reproduced from Batov et al. [54] with kind permission of the Nature Publishing Group.(f) STORM imaging of microtubules. Reproduced from Bates et al. [55] with kind permission of The American Association for the Advancement of Science.

thesis, we are mainly concerned with images as obtained by measuring beam intensity variations caused by electron scattering. Using an electron beam instead of light has a major advantage: The wavelength of electrons depends on their velocity and can be reduced to be much smaller than the wavelength of light. The resolution of an electron microscope can be down to 1 Å.

The minimum and maximum thickness of samples imaged with TEM is due to many constraints. Electrons interact with the material depending on their velocity. Electrons with a high velocity interfere less with a sample and consequently, less electrons are scattered at high velocity (Egerton [3]). Therefore, the contrast of the observed structure is low in this case. Because the velocity influences the wavelength of the electrons, there is a trade-off between resolution and contrast of the acquired image. Consequently, at a fixed velocity, samples must be thin enough to let enough electrons pass but also be thick enough to allow any scattering at all. Another constraint is the blur caused by scattered electrons that reach the camera. While electrons must be scattered for observing structure in the sample, there is also a trade-off between scattering and blurriness of the image. A detailed discussion of parameters influencing the optimal thickness is presented by Egerton [3]. Zang et al. [62] measured that an optimal result can be obtained if the sample thickness is three times the average length that an electron travels in material at a certain velocity before being scattered (mean free path length). Currently, in biological research usually samples with a maximal thickness of 300 nm are imaged (Frank et al. [63], McIntosh [64]).

Thicker samples have to be cut into thin sections to allow TEM imaging which requires prior fixation of the sample. Imaging life organisms even if their size is optimal for imaging, is not possible with current techniques due to the damage caused by the electron beam (Egerton [3]) but

also because samples must be kept at a vacuum for imaging. A natural implication is that electron microscopy can only provide insight into a single snapshot in time of any specimen that is imaged.

Ruska and Knoll [65] developed the first electron microscope in the 1930 s. This achievement gained them the Nobel price in physics in 1986. Figure 2.4(a) shows the first image of a biological sample that was imaged with their microscope. Their technology was rapidly developed further and proved an excellent tool for imaging microtubules since tubulin can be stained with heavy metals such as lead and uranyl acetate, both of which are electron dense materials and therefore easy to detect with an electron beam. The technique allowed de Harven and Berhardt [66] in 1956 to discover that microtubules are hollow (Figure 2.4(b)) which lead Slautterback [67] in 1963 to name them 'microtubules' (Bajer and Molè-Bajer [38]). A breakthrough in electron microscopy imaging (Inoue [40]) was the development of the protocol for fixation of samples with glutaraldehyde and osmium tetroxide (Sabatini et al. [68]) and later the development of fixation through freezing. The latter best preserves the natural state of a cell while chemical fixation causes more artifacts (see Section 2.2.3, McDonald [69]).

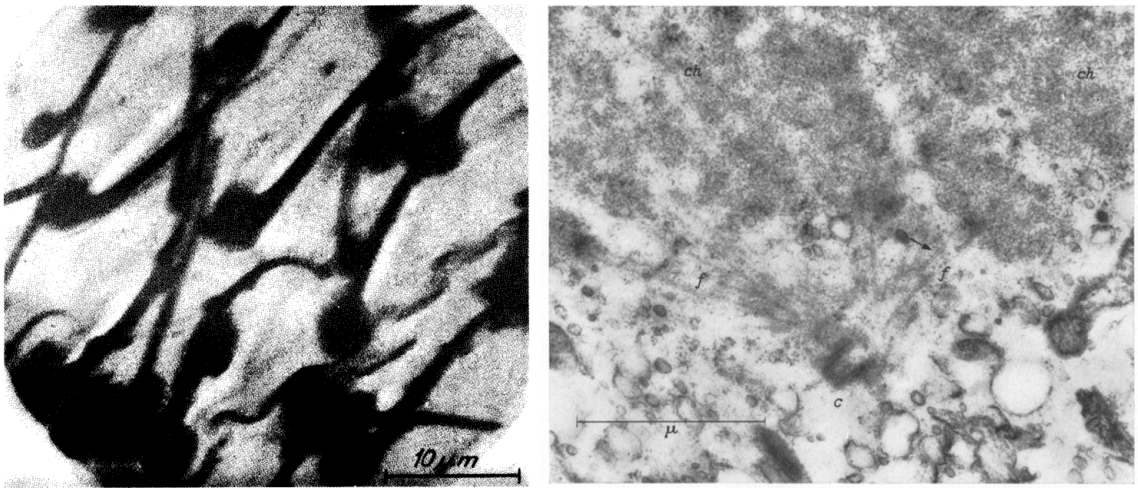
Two major techniques are currently available for imaging cells with electron microscopy: 1) Cryo-electron microscopy, which was developed in the 1980 s (see Taylor and Glaeser [70]) and 2) plastic-embedding. While the microscope is the same, these two techniques differ in sample preparation: Samples are high pressure frozen for cryo-electron microscopy and the frozen state is preserved while imaging. However, frozen samples are much harder to section than plastic-embedded samples (see Vanhecke et al. [71], Taylor and Glaeser [70], Marko et al. [72]). Therefore, cryo-electron microscopy is currently primarily used to either imaging of thin samples like small bacteria, viruses or flat parts of a cell such as the filopodium or for imaging of single sections in contrast to analyzing full stacks. Despite these difficulties, cryo-electron microscopy is widely used because the freezing with no further adding of chemicals preserves the natural state of a sample as good as it gets.

For plastic-embedding, samples are embedded in epoxy resin or metacrylate resin. If samples are fixed by rapid freezing, they have to undergo a freeze substitution of water with a substitution cocktail first (see Buser and Walther [73]). The major advantage of this technique is that, unlike frozen samples, plastic-embedded samples are comparatively easy to section with a microtome. 3D reconstructions of full cells can be obtained by stacking the thin sections after imaging. Note that aligning the sections after imaging is not a trivial task (see Saalfeld et al. [74]). Furthermore, the effective resolution of the stack in the stacking direction is limited to the section thickness which at minimum 40 nm with state-of-the-art microtomes (Saalfeld et al. [74]). To gain a better resolution in z, a tomographic reconstruction of sections can be computed. This technique, called electron tomography, will be described next.

Electron tomography DeRosier and Klug [77] were the first who used electron microscopy to gather a 3D volumetric reconstruction. To obtain a volumetric reconstruction, samples are rotated about one axis in the microscope and projections are acquired at different tilts. Figure 2.5 illustrates the basic principle. Figure 2.6 shows an example of such projections of a real specimen taken from different angles. Carazo et al. [78] formulate the problem of 3D reconstruction of the volume from these projections as such: *'Given a collection of projections (2D) g, determine the 3d structure f that produced the images g.'* Johann Radon was probably the first (Rademacher [79]) who described how to compute projections of volumes and also a solution to the inverse problem, the *inverse Radon transform* [80], in 1917. Since then, many algorithms have been developed for the reconstruction of volumes from projections.A detailed description of reconstruction algorithms and performance comparisons with a focus on electron tomography is in Carazo et al. [78], Rademacher [79] and Zeitler [81].

The problem of reconstruction, as is hard to solve already, is further complicated by the fact that 1) images suffer from noise as all measurements do and 2) the projections are incomplete. The second complication requires a little more attention. First of all, when tilting the specimens the tilt angles can only sample the full range of 180°. Second, as mentioned before, the electron beam cannot penetrate thick sections. A tilt of the specimen naturally increases the thickness of the sample as observed by the electron beam. This effect is illustrated in Figure 2.5(b) and 2.5(c). As a result, at high tilts (from about 70 degree on), no projections can be taken (see Figure 2.6(c)).

To understand the implications of missing projections, we must first understand how projections and image signal correspond. The central slice theorem states that the Fourier transform F of a



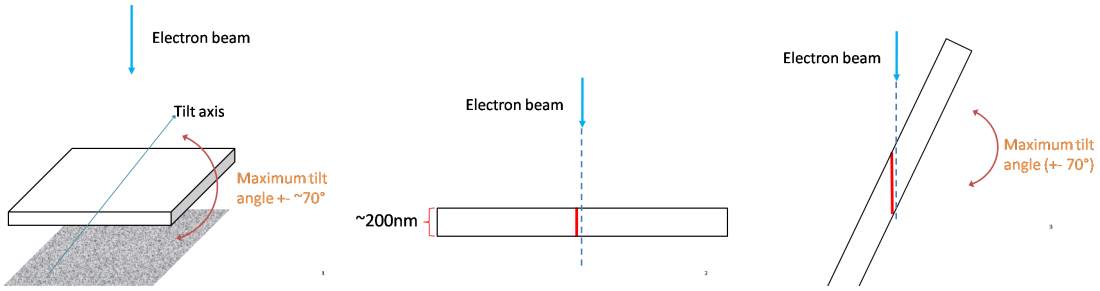
(a) (b)

Figure 2.4. (a) The first published image taken from a biological sample with a transmission electron microscope (Müller and Driest [75]). The image shows the wing of house fly. Reproduced from Fränksmyr and Ekspång et al. [76]. (b) First electron microscopic image showing that microtubules are hollow (example indicated by a black arrow). Reproduced from de Harven and Berhardt et al. [66] with kind permission of Springer Science+Business Media.

projection p from a 3D volume v is the 2D cross Section of $F(v)$ passing through the origin with the plane normal being the projection direction (see Penczek [82] and references therein). As an example, Figure 2.7(a) shows a volume that contains a tube. Figure 2.7(d) shows the amplitude of the Fourier transform of this volume. Taking projections of this volume at a 5° angle and reconstructing the volume using weighted back projection results in a slightly altered cylinder, as depicted in Figure 2.7(b). Note that reconstruction artifacts such as the horizontal dark stripe around the cylinder in Figure 2.7(b) might vary depending on the reconstruction algorithm used. Figure 2.7(e) shows the amplitude of the Fourier transform of this reconstructed volume. The original signal in Figure 2.7(d) is sampled along planes, each of which corresponds to one projection image that the volume was reconstructed from.

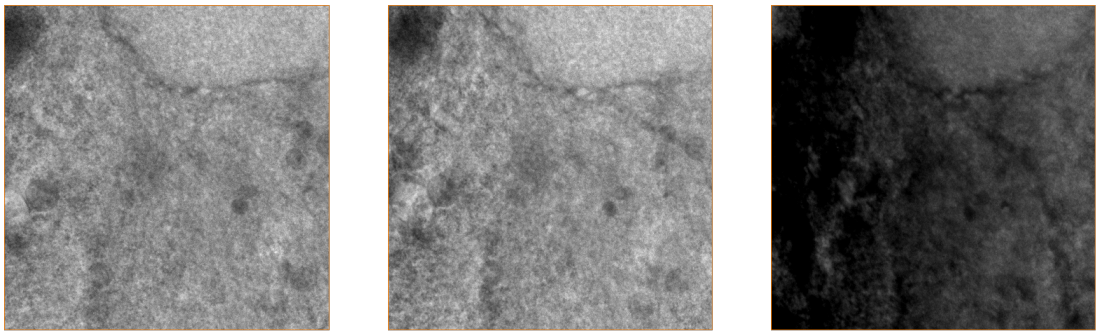
In electron tomography, not only the signal is sampled but at the same time a range of projection angles is missing completely. Figure 2.7(c) shows the same volume reconstructed from an incomplete tilt series and Figure 2.7(f) shows amplitude of the Fourier transform of the reconstruction. As a result of the missing signal, the resolution in the reconstructed volume is decreased in z-direction (see Penczek and Frank [84]) and structures appear to be smeared in the reconstructed volumes. This artifact is called the *missing wedge* and its impact on the segmentation of microtubule centerlines will be discussed in more detail in Section 3.3.1. To improve tomogram quality, specimens are often tilted around two axis and two tomograms are computed which then have to be combined. The missing information is thereby reduced to a pyramid. This approach has been first proposed by Taylor et al. [85] and has become a popular technique for imaging of plastic-embedded sections.

Reconstructing a volume from electron microscopic tilt series is challenging because for a proper reconstruction, the exact projection matrix for each projection must be known. This is not always the case, because the tilting of the sample can cause slight shifts. To correct for such alterations, often nanometric gold particles with a diameter of ~10 nm, depending on the sought magnification, are placed on the sample and projections are aligned using these particles as markers. Matching markers in different projections automatically is a challenging task (see Amat et al. [86]). For this reason, tomographic reconstruction in electron microscopy is usually a semi-automatic task. The most widely used software for creating a tomographic reconstruction from double tilt series is IMOD (Mastronarde [87]). IMOD offers several options for projection alignment: a semi-automatic particle tracking, automatic marker alignment (Amat et al. [86]) and also a fiducial-less alignment



(a) (b) (c)

Figure 2.5. (a) Principle of electron tomography: The sample is placed in the electron beam. After a projection is acquired, the sample is tilted in the microscope prior to taking the next projection. The sample thickness in the electron beam is higher at high tilt (b) than without tilting (c).



(a) (b) (c)

Figure 2.6. Projections of a sample acquired with a transmission electron microscope. (a), (b) and (c) show projections at different tilt angles.

similar to the one described by Castaño-Díez et al. [88].

2.2.3 Defects in electron tomograms

Preparing sample sections for electron tomography requires following sophisticated protocols and expertise in handling the microscope. It is hard and sometimes even impossible to guarantee quality standards for each step. Consequently, the quality of resulting tomograms is often subject to large variations. Furthermore, some effects are immanent to the technique and can never be avoided. In this section, we describe the most common artifacts and defects that we observed in electron tomograms. We only discuss defects caused by material interaction and the tomography reconstruction process and omit all optical effects such as abberation and astigmatism. See Egerton [3] for details on optical artifacts.

Fixation We observed several defects in samples which we related to high pressure freezing. Figure 2.10(c) shows microtubules that are broken by, what we believe is, a small ice crystal. Microtubules appear squished in some samples as depicted in Figure 2.10(d), an artifact we also relate to freezing. Chemical fixation on the other hand alters the structure of organelles in the sample as depicted in Figure 2.10(e) (see Graham and Beveridge [89]).

Sectioning Sectioning compresses the sample (see Studer and Gnaegi [90], Saalfeld et al. [74]). To the best of our knowledge, this effect has been measured for thin sections but not for sections with a thickness of about 300 nm. Furthermore, while sectioning samples and imaging, material

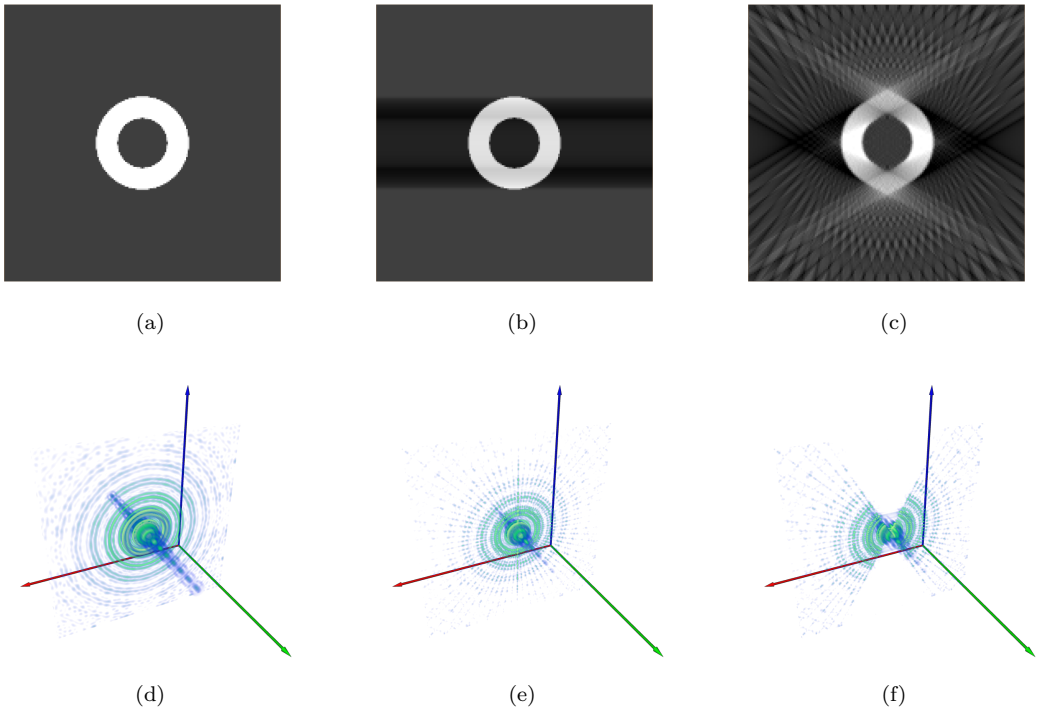


Figure 2.7. Figures (a)-(c) show cross the section of an (a) idealized cylinder, (b) a volume that was computed from 5 degree angle projections of the cylinder in (a) and (c) a volume that was computed from incomplete tilt series with 60 degree of projections missing. Figures (d)-(f) show the amplitude of the Fourier transform of the volumes (a)-(c). Backprojection was implemented as described in Schubert [83].

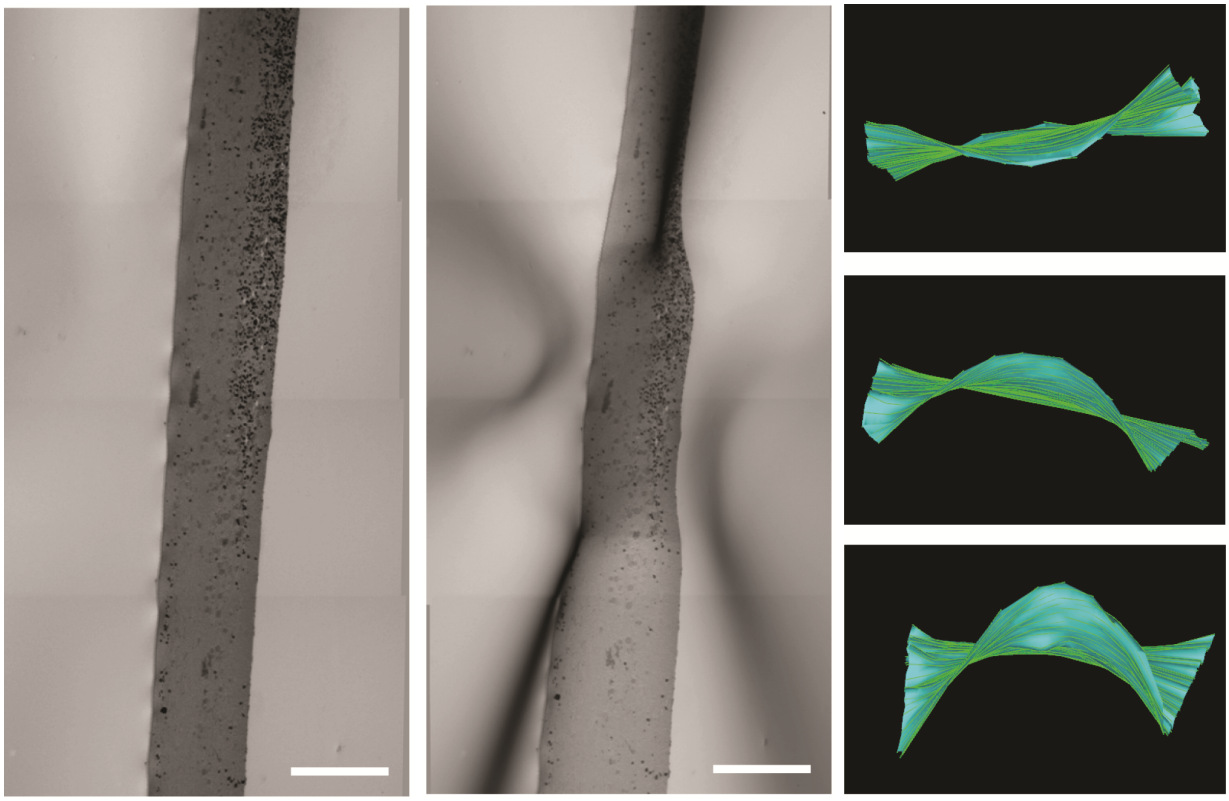
between sections is lost. The gap was been observed by several authors in the past (Moore et al. [91], Peachey [92], van Marle et al. [93], Luther [94]), however, there are no studies explicitly measuring the effect.

Staining artifacts Plastic embedded sections are often stained with heavy metal such as uranyl acetate and lead before imaging. Uranyl acetate can crystallize in the sample which results in dark spots in the volume as depicted in Figure 2.10(f). Furthermore, the stain has to penetrate the full section. If this is not the case, the interior of the reconstructed volume will have too little contrast for allowing identification of organelles in the sample. Figure 2.11(a) shows top and middle section of a tomogram where staining was not applied correctly. Figure 2.11(b) shows an x,z-slice of the volume. The contrast in the middle of the tomogram is insufficient to draw further conclusions from it.

Imaging The electron beam damages the sample while imaging. The major effect is a severe anisotropic material loss of the sample (see Luther et al. [95], [63]). This effect can cause thin sections to wrinkle if the exposure time is large. The resulting damage for a section taken from an *X. laevis* sample is shown in Figure 2.8(b). Figure 2.8(c) shows the manually drawn surface marking the outline of information contained in the reconstructed tomogram. Here, the IMOD software (Kremer and Mastronarde [96], Mastronarde [87]) was used for reconstruction. IMOD uses the outline surfaces to flatten the tomogram, but the program cannot perfectly compensate for the error. The resulting tomograms suffer from non-linear deformation.

For the imaging of large areas, the stage of the electron microscope has to be physically moved to screen the desired area of the sample because at a magnification required for imaging of individual microtubules the field-of-view of the camera is limited to a few micrometer. The sample must be imaged in tiles. Figure 2.9 shows the tiles needed for imaging a large section. The imaged patches

then have to be stitched to reconstruct the full tomogram. This task is not trivial. Software like IMOD does an excellent job joining the patches but must necessarily make certain assumptions facing elastic deformation that might not reflect the true distortion of the sample. This step might thus introduce further disturbances.



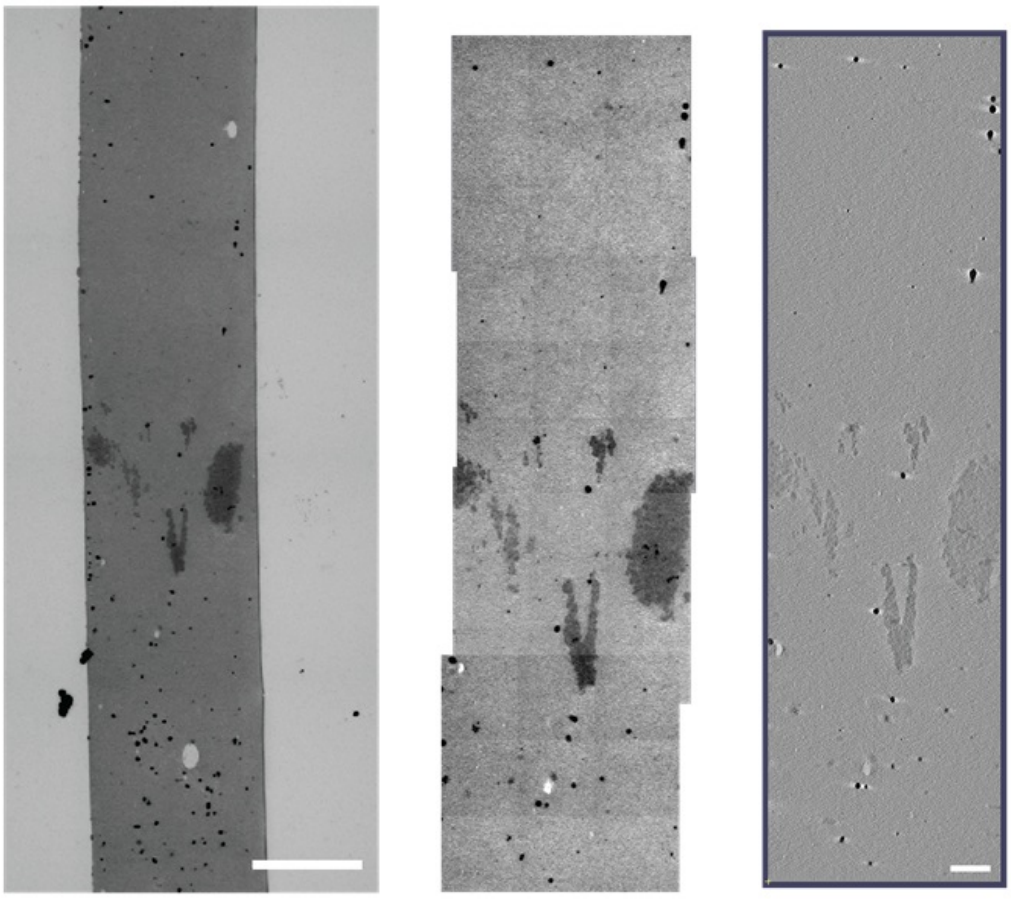
(a) (b) (c)

Figure 2.8. Illustration of warping of a section caused by exposure to the electron beam. (a) shows a low resolution image of an *X. laevis* sample before acquisition (scale bar=10 μm). After an exposure time of 11 hours, which is necessary for multiple area acquisition, the sample in Figure (a) has suffered damage as depicted in Figure (b) (scale bar=10 μm). Figure (c) shows the outline surface that marks parts of the reconstructed volume actually containing information. The surface was manually drawn using IMOD [96]. The data for the top image was acquired first and consequently has the least amount of damage. The data for the bottom image was acquired third (last) and consequently has the most amount of beam damage. Image courtesy of Erin Tranfield, EMBL.

Gold particles and reconstruction artifacts Reconstructing the volume from the projections requires aligning them first. A coarse alignment is often achieved by using gold particles as landmarks. Tracking a gold particle through all projections is challenging, because the gold particles' visibility depends on the underlying structure in the sample. At the same time, each gold particle must be aligned exactly, because errors cause structures to be smeared out in the reconstructed tomogram.

For double tilt acquisitions, the volumes for each two axes have to be combined. Typically, again gold particles are used for an initial alignment of the two volumes. This step is as crucial as the gold particle tracking before: For example, an offset of 20 nm only will cause each microtubule to appear twice in the tomogram with half its signal, as depicted in Figure 2.11(c) and 2.11(d).

If too few gold particles are placed on top of the sample, the alignment algorithms might not have enough features for alignment. Too many gold particles on the other hand might make



(a) (b) (c)

Figure 2.9. Illustration of the patch-wise imaging of large samples. (a) shows a low resolution image of an *X. laevis* sample before acquisition (scale bar=10 μm). (b) illustrates the patches needed for imaging. Here, a 3×3 block (montage) can be acquired by moving the electron beam. To image the full sample, 4 overlapping 3×3 blocks have to be acquired by moving the microscope stage. To make a stitching of these blocks feasible, they have to overlap. (c) shows a slice of the final reconstructed, flattened and stitched tomogram (scale bar=2 μm). Image courtesy of Erin Tranfield, EMBL Heidelberg.

tracking individual particles hard or even block the electron beam completely.

Dirt Because the magnification allows resolving structures at a nanometer scale, even a single particle of dust, will be largely visible in the final image. The alignment of the projections is hard to perform for a dirty sample, because gold particles can be hidden by the dirt. A fiducial-less alignment based on the image data is also impossible in such a case. In the worst case, the electron beam is blocked and the signal of the image is too low to allow further processing. Figure 2.11(e) shows an example of a projection of a dirty sample and 2.11(f) a slice in the corresponding tomogram that was reconstructed from the projections. It is hard to clean dirty sections and therefore these sections are often useless.

2.3 Model organisms

Life is complex and the amount of knowledge a single researcher can gain in a lifetime about one organism is limited. For this reason, in molecular biology there is a general consent that molecular

processes are studied only in few organisms before moving on to the full spectrum of organisms available on earth. These selected organisms are referred to as *model organisms* (Alberts et al. [1]). Model organisms were chosen for natural properties of the organism but also for their ease of use. For example, a transparent organism is easier to study with a light microscope than an opaque organism. Short development cycles of the organism are essential for conducting experiments. Also, genetic properties of an organism such as the ease of creating mutants are selection criterion. It's size matters as well.

Most samples used in this thesis come from early embryos of *Caenorhabditis elegans* (*C. elegans*). *C. elegans* is a small transparent nematode, only 1mm in length. The worm has a fixed number of cells and their development follows an exact pattern, that is, each cell division results in two cells the function of which are known. This known *cell lineage* make *C. elegans* an interesting model for research on cell function in multicellular organisms. *C. elegans* was the first multicellular organism to have its complete genome sequence determined (Alberts et al. [1]). Furthermore, genetic manipulation such as RNAi can be easily achieved by feeding the worm bacteria expressing a particular RNA (Hammell et al. [97]). Due to these properties, *C. elegans* is an excellent organism to study microtubules. See Alberts et al. [1] for more information.

Xenopus laevis (*X. laevis*), the African claw frog, is also useful to study spindle development. Spindles can be created in purified egg extract of these frogs which allows a clear view on the spindle assembly undisturbed by other organelles (see Karsenti and Vernos [11] for a review). However, the meiotic spindle of *X. laevis* is about 25 μm long which makes imaging of a full spindle particularly challenging due to the limited field of view of an electron microscope at the required resolution.

We will furthermore show examples from HeLa cells, an immortal human tumor cell line and epithelial cells of *Drosophila melongaster* (*D. melongaster*), the common fruit fly. As a proof of concept we will furthermore evaluate the effectiveness of our algorithm on samples of *Trypanosoma brucei* (*T. brucei*), a single cell organism that causes the sleeping disease. This organism is mostly studied for the function of the flagellum used for movement but we evaluate the algorithms performance on the outer skeleton formed by uniformly spaced microtubules, that stabilizes the cell membrane for *T. brucei* (Farr and Gull [22], Gull [23]).

The details on the sample preparation protocols and electron microscopy parameters will be described in the following chapters separately when needed.

2.4 Discussion

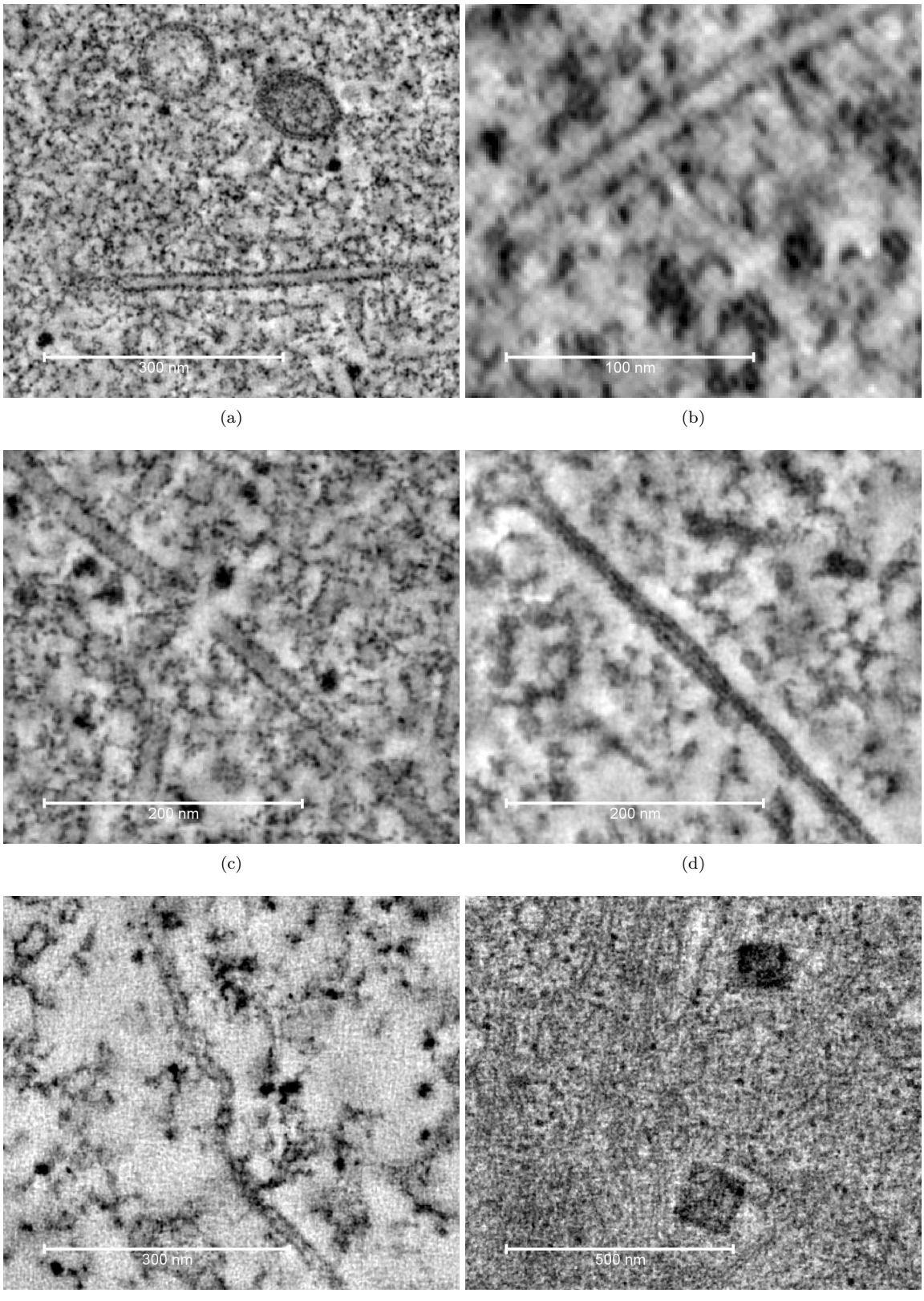
Preparing samples for electron tomography and imaging is not only time consuming but also a delicate matter. It can take a biologis years to gather sufficient experience in the protocol. Apart from the described effects, each of the steps (freezing, cutting, mounting, imaging) is likely to cause damage to the very small samples, in particular, to the 300 nm thin sections after microtomy. Success in preparing a perfect sample can also depend on environmental conditions: For example, even light vibrations in the vicinity of the microtome can disturb the accuracy of the sectioning. Sections in this case might be too thick for staining (Figure 2.11(a) and 2.11(b)), become unusable and consequently will be missing from the stack of sections.

In our experience, approximately one out of ten sections prepared for electron tomography are either not usable or destroyed completely. This is problematic, because if one section is lost, the full sample is lost since a 3D reconstruction cannot be obtained from an incomplete stack of sections. Furthermore, many defects can be detected only after the section is reconstructed. Upon detection of a lost section, often, much time has already been wasted on the preparation of different sections from the same sample. For these reasons, depending on the size of the structure of interest, it can take an experienced biologist several months to obtain a complete set of tomograms that cover the full sample.

The variety of image modalities, as described in Section 2.2.3, makes image processing of the electron tomograms particularly challenging. Resolution and contrast are strongly influenced by many conditions and a segmentation algorithm must be capable of robustly producing accurate results despite these variations. In addition, tomograms contain a variety of biological structures apart from microtubules, such as vesicles, membranes, mitochondria, ribosomes and DNA. Any algorithm for microtubule detection must be able to discriminate these structures from micro-

tubules, which is not trivial since they may locally appear similar to a tubular shape. Finally, it is yet unclear how much the missing wedge influences the validity of the results. The missing wedge diminishes the signal of microtubules and render some close to invisible. Therefore, it is not immediately evident that electron tomography is suitable for the analysis of microtubule networks at all.

All of these effects also have an impact on the stitching on subsequent tomograms. Even for an expert it is already hard to identify matching patterns in top and bottom slices of a tomogram that would allow to stitch the data. Any automation based on the image data would have to deal with foreign matter, gold particles, contrast differences and often the lack of significant features that can be used for stitching.



(a) (b)

(c) (d)

(e) (f)

Figure 2.10. (a) and (b) Slices through an electron tomogram with a reasonable quality. (c) Tomogram with frequent breaks in the structures. We suspect this is a freezing artifact. (d) Microtubules which are squashed. Again, we suspect this is a freezing artifact. (e) Bends in microtubules caused by chemical fixation.(f) Osmium crystals in a sample.

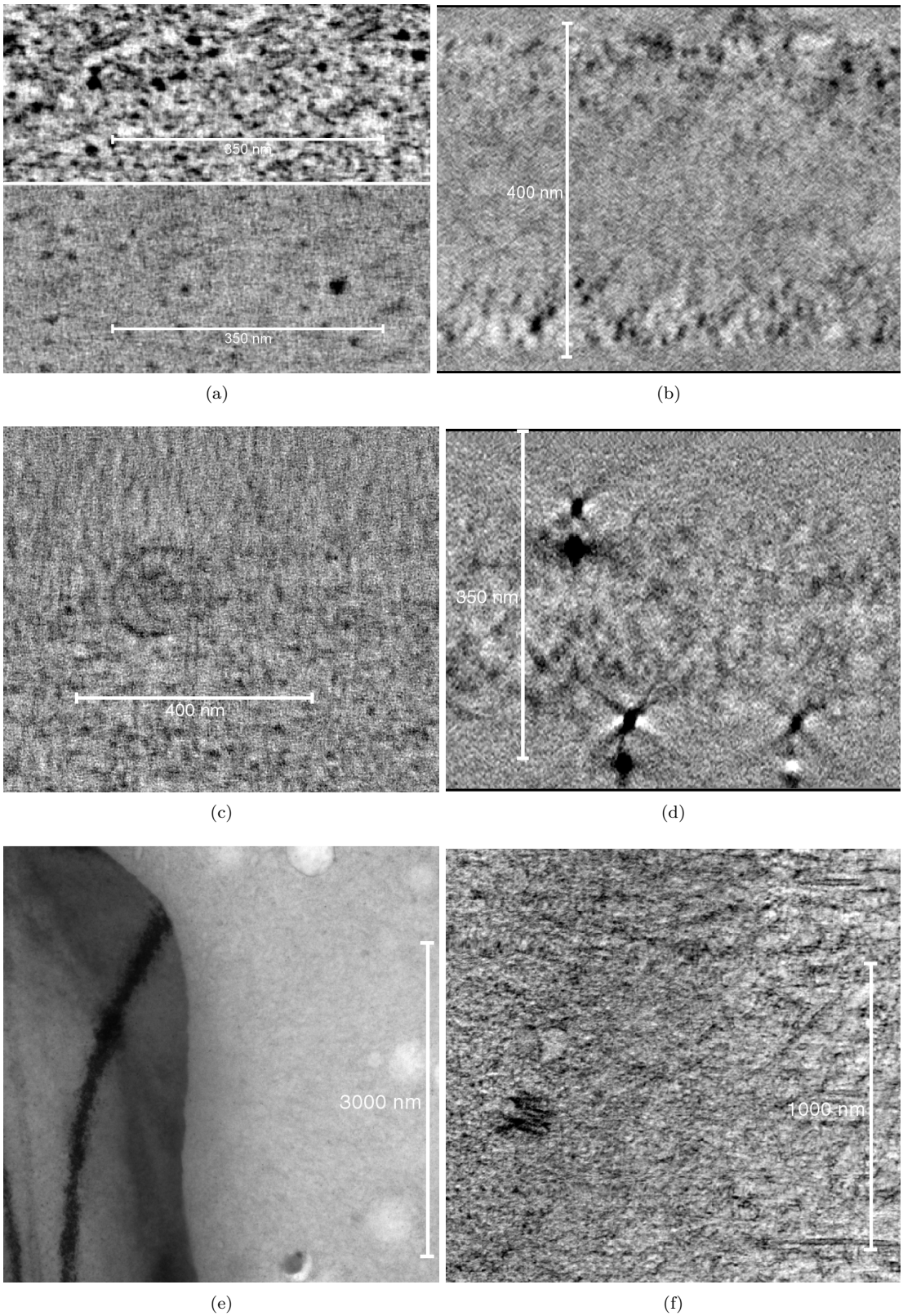


Figure 2.11. (a) and (b) Slices from tomogram with anisotropic staining. (c) and (d) Erroneous tomogram reconstruction due to a gold particle mismatch at the double tilt alignment. Gold particles appear twice (d).(e) and (f) Projection and tomogram of a sample with a drop of fluid of unknown origin on it. (a)-(b) Data courtesy of Ina Woog, MTZ Dresden (c)-(d) data courtesy of Stefanie Redemann, MTZ Dresden.

Chapter 3

Segmentation of microtubule centerlines in electron tomograms

In the previous chapter, we discussed the usefulness of electron tomography for the study of microtubule organization. In order to quantify any properties of microtubules in a sample, the microtubules have to be first identified and converted to a representation that can be analyzed. In this chapter, we discuss the detection of the centerlines of microtubules and their representation as polygonal lines. We use the terms *line* and *polygonal line* equivalently. We call the extraction of centerlines *segmentation* even though this term usually refers to a partitioning of an image into several regions associated with particular objects. The resulting set of polygonal lines representing the microtubule centerlines will be referred to as a *tracing*.

Segmentation of microtubules has so far always been a manual process due to the lack of image processing software designed for the task. IMOD (Kremer et al., [96]) is probably the most popular software package used for tomographic reconstruction and 3D manual segmentation of cellular features. Examples in which IMOD was used for microtubule segmentation are the analysis of the organization of interphase microtubules in fission yeast (Hoog et al. [5]), analysis of the meiotic spindle assembly in *C. elegans* (for example, Srayko et al. [6], O'Toole et al. [7]), or the 'visible cell project', whose goal is to create a full segmentation of an entire mammalian insulin secreting pancreatic beta cell (Marsh and Mastronarde [98], Marsh [99]). In all the studies, organelles were segmented manually in a small number of biological samples.

Unfortunately, manual segmentation of microtubules is very time consuming (see Weber et al. [4]) and it is currently the major limitation when aiming at a statistical analysis of large microtubule networks. For this reason, biological research on microtubules using electron tomography is currently often based on single observations as opposed to deriving statistics from many samples.

In this chapter, we describe our efforts to facilitate the segmentation of microtubule centerlines for a large number of electron tomograms. First, we describe a fully automatic segmentation method to extract the centerlines. We apply template matching to first enhance the microtubules in the tomogram (Weber et al. [100], Rigort et al. [101]). We describe the basics for this approach in Section 3.1.1 and our implementation in Section 3.3.1. We propose to segment microtubule centerlines with the method described by Rigort et al. [101] which we modify to adjust for the different modalities of cryo-electron tomography and electron tomography of plastic embedded samples (Section 3.3.2).

To enable the biologist to quickly check success or failure of the method, apply necessary corrections to the results and eventually classify microtubule's end morphologies, we developed an editor designed to fit the specific needs for this task. This editor is easy to use and meets the speed requirements of large-scale applications (Section 3.4).

The main part of this chapter, Section 3.5, describes how we measured the performance of the tracing algorithm. We conduct a detailed evaluation of the proposed method on a large number of single- and dual-axis tomograms obtained from central sections through centrosomes in early *C. elegans* embryos during mitotic spindle assembly. We first analyze the sensitivity of the tracing parameters and then compare the quality of automatically traced microtubules to manual tracings. Furthermore, we show that the automatic approach is essential to obtain reliable results because

manual segmentations are subject to large variations and results can therefore be strongly biased.

The automatic tracing yields 4% false negatives and 8% false positives on average for dual axis tomograms. For single-axis tomograms, the accuracy of tracing is lower (16% false negatives and 14% false positives) due to the missing wedge in electron tomography. In addition, we show that the algorithm is applicable to biological samples from different species (human HeLa cells, Drosophila wing) and to samples prepared with different preparation methods (chemical fixation, high-pressure freezing).

In summary, our main contributions are:

- We show how microtubules can be reliably segmented in electron tomograms with template matching.
- We evaluate the method in detail on a large quantity of true biological data from different samples and with different image modalities.

Large parts of this chapter, such as the previous introduction, are copied from our publication '*Automated tracing of microtubules in electron tomograms of plastic embedded samples of Caenorhabditis elegans embryos*' [100]. The copied passages are not explicitly annotated. The method described here was similarly published in '*Automated segmentation of electron tomograms for a quantitative description of actin filament networks*' [101]. Our main contributions to this paper were the missing wedge adjustment for the template and the evaluation of the result. We described preliminary efforts for microtubule tracing and the interactive editor in our publication '*Fast Tracing of Microtubule Centerlines in Electron Tomograms*' [4].

3.1 Previous work on finding line-like structures in electron tomograms

3.1.1 Template matching

A popular approach for the segmentation of structures in electron tomograms with known shape is template matching (see Förster and Hegerl [102]). Here, small volumes, containing an idealized representation of the sought shape are computed. These volumes are called *templates*. Then, the similarity of volume and template is computed at each pixel. Figure 3.1 illustrates this procedure for a 2D image. The template acts as a mask that is shifted along the image. At each point, the similarity of mask and image below the mask is computed.

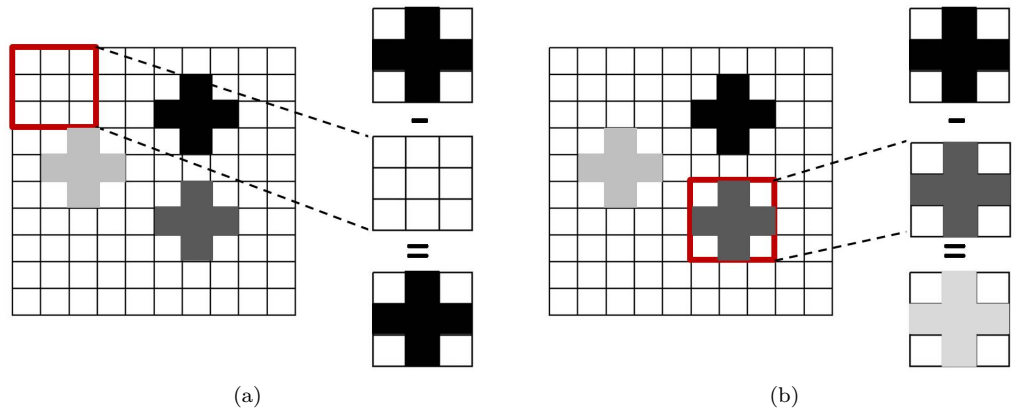


Figure 3.1. Illustration of the general principle of template matching. An image is scanned for areas similar in pixel value to the template containing a cross. (a) The image part enclosed in the red square does not match the template. The difference image still contains the cross. (b) Image enclosed in red square and template are similar. The difference image is nearly empty. The difference in contrast can be compensated for by normalizing template and image under the mask.

A common similarity measure is the normalized cross-correlation of image and template. Both the image under the template mask I and the template t are normalized so that the variance of resulting image and template is one and the mean is zero. The normalized cross-correlation is derived from the quadratic difference of the normalized image pixels as follows:

$$f(I,t) = \sum_{i,j=1}^{M,N} \left(\frac{I(i,j) - \mu_I}{\sigma_I} - \frac{t(i,j) - \mu_t}{\sigma_t} \right)^2,$$

where I and t are both $N \times M$-dimensional images with standard deviation σ_I, σ_t and mean μ_I, μ_t of the pixel values. $t(i,j)$ extracts the pixel at position i,j (likewise for I). Eliminating the brackets, this equation can be written as

$$f(I,t) = \frac{1}{\sigma_I^2} \sum_{i,j=1}^{M,N} (I(i,j) - \mu_I)^2 + \frac{1}{\sigma_t^2} \sum_{i,j=1}^{M,N} (t(i,j) - \mu_t)^2 - 2 \sum_{i,j=1}^{M,N} \frac{(I(i,j) - \mu_I)(t(i,j) - \mu_t)}{\sigma_I \sigma_t}$$

The first two terms evaluate to 1. It suffices to evaluate the term

$$\mathrm{C}(I,t) = \sum_{i,j=1}^{M,N} \frac{(I(i,j) - \mu_I)(t(i,j) - \mu_t)}{\sigma_I \sigma_t}$$

to measure the difference between image and template. Because of the normalization, C always lies between -1 and 1.

The computation of the normalized cross-correlation for a template with an arbitrary mask at each pixel of an image can be computed with three convolutions of image and template (see, for example, Roseman [103], Padfield [104]). This property makes the technique applicable to larger datasets because a convolution can be efficiently computed by a multiplication in the Fourier domain. Furthermore, the normalization of image under the mask and template compensates for contrast variations in the image. These properties make cross-correlation computation a popular tool for various tasks in image processing of electron tomograms.

A common application of template matching on electron tomograms is particle picking (for an overview see: Zhu et al. [105]). Template matching is also used for the enhancement of other structures. Lebbink et al. [106], for example, applied template matching to automate the identification of membranes and microtubules, but they give no explicit segmentation algorithm resulting in a set of microtubule centerlines. They concluded that template matching can help annotating tomograms, but manual corrections are required. Cross-correlation is a common tool for computing image alignments (see, for example, Kremer et al. [96]). Recently, the company FEI developed a software (Schoenmakers et al. [107]) that can be used for template matching in electron tomograms. Finally, in Rigort et al. [101], we showed how template matching combined with an automated tracing algorithm can be used to trace actin filaments in cryo-electron tomograms in a fully automated fashion. We describe the details of our implementation of template matching for electron tomograms in Section 3.3.1.

3.1.2 Methods for detecting tubular structures

Most closely related to the problem of detecting microtubules is the research on vessel segmentation in medical imaging. Schaap et al. [108] give an overview of state-of-the-art methods including a detailed evaluation of the effectiveness of different methods. However, none of these methods can be directly applied to the extraction of microtubules in electron tomograms, because the image data differs drastically between medical imaging and electron tomography. This problem is described by Sandberg [109], who gives an overview of methods for structure enhancement and segmentation methods applied to electron tomograms, including one for line-like structures. The emphasis of [109], however, is on a description of the challenges faced when trying to fully automatically segment objects in electron tomograms.

To support the manual tracing of line-like structures, several interactive tools have been developed. For example, Glaser and Glaser [110] developed an editor for neuron reconstruction. Smith et al. [111] describe a tool for semi-automatic actin segmentation in light microscopy images. Foe

electron tomography, the most widely used tool for tracing line-like structures is IMOD (Kremer and Mastronarde [96]).

The literature describing solutions for the automatic tracing of line-like structures in electron tomograms is rare. Sandberg and Brega [112] describe a method called the line filter transform which is designed to enhance line-like structures in images. Although their method is promising, it is not readily applicable to 3D volumes. Jiang et al. [113] present a model-based approach to extract microtubules and segmentation of the structure of the ends. Their approach, however, seems to be rarely used in practice and their software does not seem to be available to the public. Only recently, Nurgaliev et al. [114] propose an ad hoc image filtering method to find extended pieces of microtubule centerlines in tomograms, which are then connected to longer centerlines in a post-processing step. To our knowledge, this is the only serious approach for the automatic extraction of microtubules in electron tomograms. The authors conclude that their approach is promising but cannot replace manual tracing by a human expert at this stage. Thus, to the best of our knowledge, fully automated and reliable tracing of microtubules in electron tomograms has not been presented so far.

3.1.3 Similarity measures for lines

Given two polygonal lines, it is not immediate obvious how their similarity can be measured. Several approaches have been discussed in the literature. Measures to evaluate the quality of a time series (see Ding et al. [115] for an overview) have been applied to analyze trajectories of organisms (Leitte et al. [116]) and to biological data such as time lapse imaging of microtubule growth (Gelasca et al. [117]). Schaap et al. [108] propose a standard for evaluation vessel tracking algorithms. A ground truth is generated from manual segmentations of three human tracers. It is generated on 32 volumes each of which contains 4 manually traced lines representing the vessel centerlines. The ground truth is generated by tracing the vessel; computing a mean centerline from the three tracings with a mean shift algorithm; correcting the tracings manually; and then again using mean shift to compute the final ground truth. Here, manually traced lines are always connected and no false positives are to be expected. Jomier et al. [118] use STAPLE to generate a ground truth for line segmentations. The ground truth is represented as a density map and tracking algorithms are evaluated by computing the mean density on the traced lines. Other error metrics based on pixel values are described in Jain et al. [119]. A sophisticated procedure to combine neuron tracings created by many different users to a ground truth represented as lines is described in Helmstaedter et al. [120]. This problem is particularly challenging, because of the complex branching of axons and dendrites.

However, in contrast to the described work, when comparing microtubule centerlines, we can make some simplifying assumptions: Microtubules do not branch and the distance of centerlines is at least 20nm. Microtubules are rather straight, so strong curvature must not be taken into account. The simplest way to compare line tracings is converting lines to point clouds and measuring the points distances (see, for example, Cole et al. [121]). This measure can give an idea on the quality of the result, but it cannot detect errors in the reconstructed geometry. However, for the specific case of single distinct lines together with a comparison of the number of lines in both sets it becomes a strong indicator for the quality of the segmentation. Another advantage of this simple comparison is that results are easy to interpret even without expert knowledge in statistics or geometry. Therefore we will use a pointwise comparisons of the lines later in this chapter to evaluate our results.

3.2 Sample preparation, imaging and tomogram reconstruction

Samples were prepared and imaged at the MPI-CBG Dresden and at the EMBL Heidelberg by our collaborators as described in the following paragraphs. We repeat the protocol description from our publication [100] as a reference for the interested biologist.

C. elegans Specimens are prepared for electron microscopy as described in Müller-Reichert et al. [122]. Early *C. elegans* (Bristol N2) embryos are sucked into capillary tubes cut to 1 mm length. The capillary tubes are placed in 100-lm-deep membrane carriers (Leica) in M9 buffer containing 20% BSA. The embryos are then frozen using an EMPACT2 (Leica) high-pressure freezer with a Rapid Transfer System (RTS). Samples are freeze-substituted in a Leica AFS2 using a freeze-substitution cocktail containing 1% osmium tetroxide and 0.1% uranyl acetate in anhydrous acetone. Specimens are warmed to room temperature, washed in fresh anhydrous acetone and gradually infiltrated in Epon/Araldite resin. After infiltration, samples are embedded in a thin layer of Epon/Araldite between microscope slides coated with Teflon®.

D. melongaster Wings from pupal Drosophila were dissected in PBS buffer, placed in 100 μm-deep membrane carriers containing hexadecene, high-pressure frozen and freeze-substituted as described above. The samples were infiltrated and thin-layer embedded in Epon LX112 resin.

Human HeLa midbodies As described previously (Guizetti et al. [123]), human HeLa cells were grown on sapphire discs in DMEM medium containing 10% SVF, 2 mM L-glutamine at 37° C in 5% CO2 and cryo-immobilized using an EMPACT2 + RTS high-pressure freezer (Leica Microsystems, Vienna, Austria). Specimens were frozen in 200 lm-deep membrane carriers (Leica) filled with growth medium containing 20% BSA (Sigma). Samples were freeze-substituted in acetone containing 1% osmium tetroxide and 0.1% uranyl acetate and embedded in Epon-Araldite.

Chemically fixed HeLa cells Hela cells were grown on Aclar coverslips, fixed for 1 h in 2% glutaraldehyde in sodium cacodylate buffer, and postfixed in 2% osmium tetroxide. Samples were dehydrated through a graded series of ethanol and thin-layer embedded in Epon-Araldite.

Sectioning and staining for electron tomography After polymerization, one glass slide is removed and samples are remounted on dummy blocks for ultramicrotomy. 300 nm thick longitudinal serial sections (Ultracut UCT Microtome, Leica) are collected on Formvar-coated EM slot grids and post-stained with 2% uranyl acetate in 70% methanol (15 min) followed by Reynold's lead citrate (5 min). For electron tomography, 10 nm colloidal gold particles are deposited to both surfaces of the sections.

Electron tomography Electron tomography is performed in a Tecnai F30 microscope (FEI, The Netherlands) operated at 300 kV, equipped with a US1000 camera (Gatan). Midbody tomograms are acquired on a different microscope (Tecnai F30, with an 4 k Eagle camera from FEI). Projection images are collected over a $\pm 60°$ range at 1° increments using the automated image capture software SerialEM (Mastronarde [124]). The pixel size during acquisition was 1.178 nm, except for the HeLa midbodies, where it was 1.5 nm. Reconstruction from the projection stacks is carried out using the IMOD software package (Kremer and Mastronarde [96]), using the gold particles as fiducial markers. The tomograms were acquired using a magnification of 9400×. Three samples were acquired a second time at a magnification of 4700× to test the performance of the tracing with lower magnification.

3.3 Reconstruction of microtubule centerlines

The automatic tracing algorithm consists of two main steps. The first step is to compute the local normalized cross-correlation of the tomogram with a template that mimics a short microtubule segment. The second step is to trace the lines using the correlation value and computed orientations as indicators for the microtubules' location.

3.3.1 Template matching for finding microtubules in electron tomograms

We use a hollow cylinder whose inner and outer radius matches a microtubule's geometry as a template. The template has a slim cylindrical background mask. Thus the template has two free

parameters: its length and mask diameter. Figure 3.2(a) shows an illustration of the template and its parameters. We compute cylinders separately for multiple orientations that uniformly sample the half sphere (about 850 orientations). The cylinders are sampled with twice the frequency that would arise from the voxel size of the tomogram and are then binned by two to avoid aliasing. We take neither the resolution of the electron tomogram nor the angular spacing of projections (see Section 2.2.2) into account when computing the template.

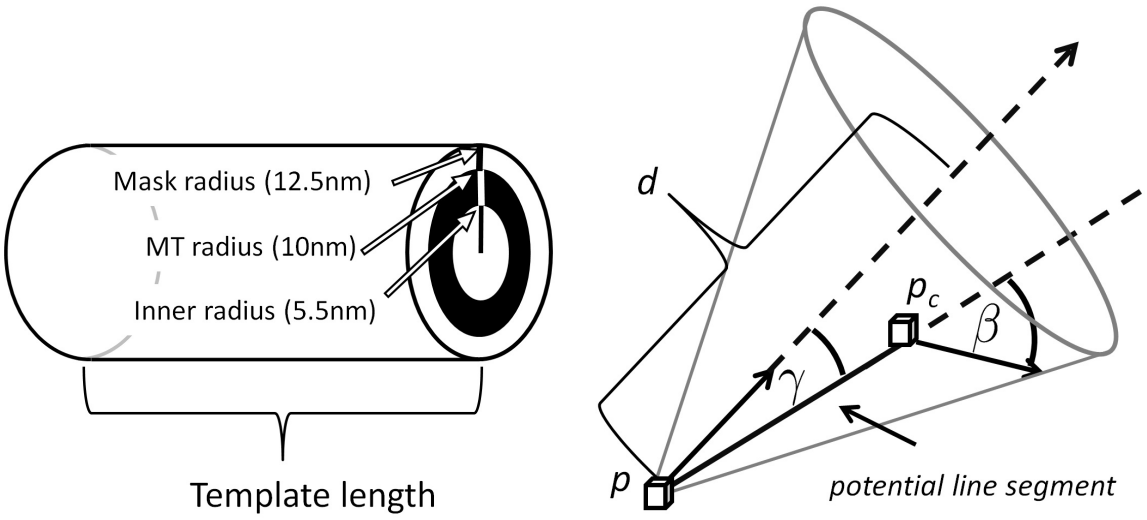


Figure 3.2. (a)Illustration of the parameters for the template tube. (b) Illustration of the algorithm used for microtubule centerline extension. Arrows: Orientations computed by template matching. γ, β: angles between potential line and orientations. d: length of search cone. p: current last point on computed line. p_c: candidate voxel in the search cone.

To compensate for the missing wedge or pyramid (see Section 2.2.2), as proposed by Förster and Hegerl [102], each template is adjusted by transforming it to the Fourier domain, removing the signal missing in the tomogram also from the template signal, and transforming it back to the spatial domain before computing the normalized cross-correlation with the tomogram. The resulting template for three orientations is shown in Figure 3.3. Its structure depends on its orientation in the tomogram (compare Figure 3.3 a, b and c), as is the case for the appearance of microtubules in electron tomograms. To compensate for boundary artifacts, we compute the template with a length of 100 nm and cut the border before normalization. We implemented the computation of cross-correlation in the Fourier domain as described by Roseman [103]. The required convolutions are computed on a GPU using CUDA [125]. For each voxel, we store the highest correlation value and the template orientation yielding this value.

3.3.2 Tracing microtubule centerlines

The second step is a line search that starts from a seed point and iteratively finds the next voxel on the centerline as described by Rigort et al. [101]. In brief, a search cone from the current voxel p is spanned in the template direction up to a length d (see Figure 3.2(b)). Each voxel in the search cone is a potential candidate for the next point on the microtubule centerline. For each candidate voxel p_c in the search cone, we compute the value

$$\mathrm{S}(p, p_c) = \mathrm{C}(p_c)\,\mathrm{L}(p, p_c)$$

where $\mathrm{C}(p_c)$ is the normalized cross-correlation at the candidate point, and $\mathrm{L}(p, p_c)$ is defined by

$$\mathrm{L}(p, p_c) = \exp(-(\beta + \gamma)/w_o^2$$

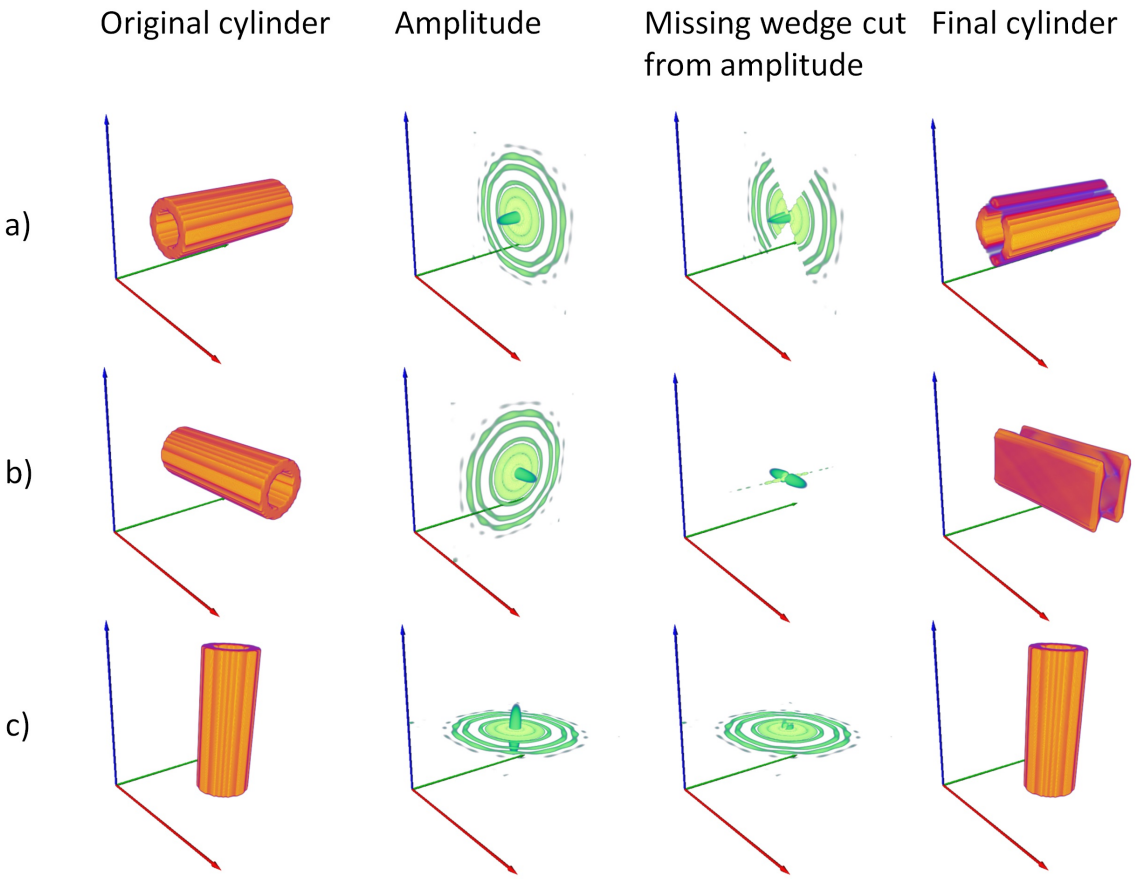


Figure 3.3. Illustration of template adjustment to the missing wedge in single-axis tomograms for computing cross-correlation. Left to right: Original cylinder template, amplitude of cylinder template signal in the frequency domain, missing wedge cut from signal, final normalized cylinder template. Rows: (a) Template is parallel to tilt axis. (b) Template is perpendicular to tilt axis in x–y-plane. (c) Template is parallel to z-axis. *Green*: Tilt axis (y-axis). *Blue*: z-axis, *Red*: x-axis.

where w_o is a weight parameter, and β and γ are the angles between vector $p - p_c$ and the template orientation computed for points p and p_c (see Figure 3.2(b)). L is similar to the smoothness term defined by Fischer and Buhmann [126], but there, it implicitly penalizes deviations of the line from the orientation field and strong curvature. The voxel yielding the highest S value is chosen as the next point. From this next point, we start the new search. For performance reasons (processing time), we begin the search for the next voxel at a distance of at least 5 nm from the current voxel. The line search stops if no p_c can be found in the search cone such that $\mathrm{S}(p, p_c)$ exceeds a threshold S_{min}. The automatic tracing uses all voxels as seed points whose correlation exceeds a fixed threshold C_{min}. Note that in contrast to Rigort et al. [101], we omit the co-circularity and distance term since we found that both have only marginal influence on the result. We also do not apply the non-local-means filter as pre-processing step as proposed by Rigort et al. [101], because we observed that filtering did not improve the results for our tomograms.

3.4 Interactive editor for manual tracing of microtubules

In order to visualize, edit and annotate the centerlines obtained from automatic segmentation, we developed an editor in Amira (see Westerhoff and Hege [127]) designed for the specific task of viewing and correcting traced microtubule centerlines quickly. Figure 3.4 shows a screenshot of the user interface. Centerlines can be searched in a slice that can be moved and rotated freely in the volume (Figure 3.4, center). A second window shows the network traced in 3D along with the

current slice orientation (Figure 3.4, right). This feature can be used, for example, to check the orientation of microtubules with respect to cellular organelles, like the centrosome. This general design is similar to the one implemented in IMOD (Kremer and Mastronarde [96]). If a microtubule is missing from the automatically computed centerlines, it can be traced in the editor using a semi-automated approach. To do so, the user places a seed point and the microtubule is traced automatically using the correlation and direction fields (see Section 3.3.1). For the visualization and annotation of microtubules, the editor can automatically unfold a selected microtubule and display a slice extracted from the tomogram around each line segment on the plane. The slice orientations for each line segment are automatically determined to minimize the angle between slice normal and z-axis, since the microtubules can be seen best in the plane perpendicular to the z-axis. For example, in Figure 3.3 (b), a slice oriented along the plane spanned by the blue and red axes would not show the microtubule at all when the missing wedge is cut out, whereas a plane through the red and green axes would. This tool allows a user to see each microtubule immediately in its full length and quickly determine if the centerline was traced correctly. The editor provides tools for corrections, such as lengthening, shortening and connecting lines. It also allows the user to quickly navigate to the ends and annotate them.

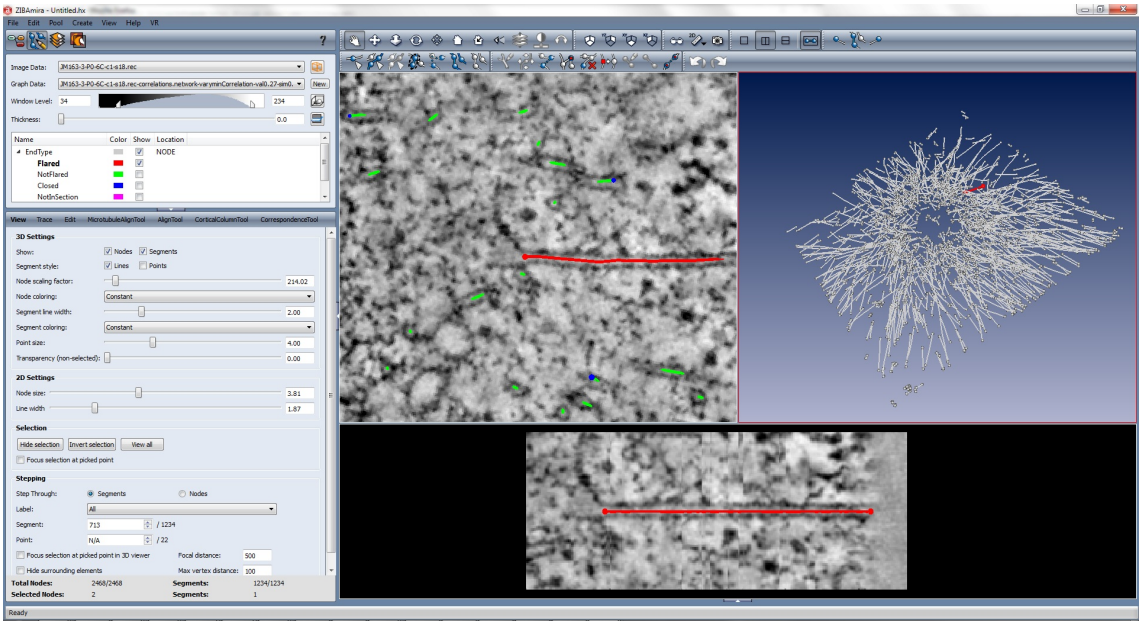


Figure 3.4. Screenshot of the line tracing editor. Left viewer (middle window in this screenshot) is used for editing lines. Right viewer displays the current segmentation in 3D. The bottom viewer shows the full line and the surrounding volume data unfolded in the plane.

3.5 Evaluation of results

In order to evaluate the quality of the automatic tracing (Section 3.3.2), we compare lines traced by experts to lines traced by the algorithm described in Section 3.5.1. As a quality measure, we compute a pointwise comparison as proposed, for example, by Cole et al. [121].

We found that experts very rarely mistake a different structure for a microtubule but tend to miss many microtubules which renders the analysis of false positives difficult. To get a reliable estimation of the quality, we generate a ground truth for microtubules from several manually segmented centerlines (Section 3.5.2). We use this ground truth to measure the sensitivity of parameters that are used by the tracing algorithm (Section 3.5.3).

3.5.1 Pointwise comparison

We identify false positives and false negatives for a tracing by establishing correspondences of points between the tracing and a reference tracing that was created manually. Lines in both sets are sampled densely (1 nm). For each point p in one line set, we search for points in the other line set that are close to p. We call p *matched*, if we find a point q in the other line set within a small radius r from p (see Figure 3.5(a) for an illustration). We define the set of false positives (FP) as all points in the tracing that were not matched (Figure 3.5(a), red dots). The set of false negatives (FN) contains points in the reference that are unmatched (Figure 3.5(a), blue dots). The set of true positives (TP) (Figure 3.5(a), green dots) contains all matched points for either the tracing or the reference. Because TP can vary slightly depending on whether TP is counted in the tracing or in the reference, we indicate this with a subscript (TP_t for the tracing and TP_r for the reference). To quantify the quality of the tracings, we define the ratio of false positives (FPR) in the tracing as

$$FPR = \frac{\#FP}{\#TP_t + \#FP}$$

and the false negative ratio (FNR) in the reference as

$$FNR = \frac{\#FN}{\#TP_r + \#FN}$$

The FNR measures the percentage of the points that are missing, while FPR measures the percentage of the points that are actually false positives. [1]

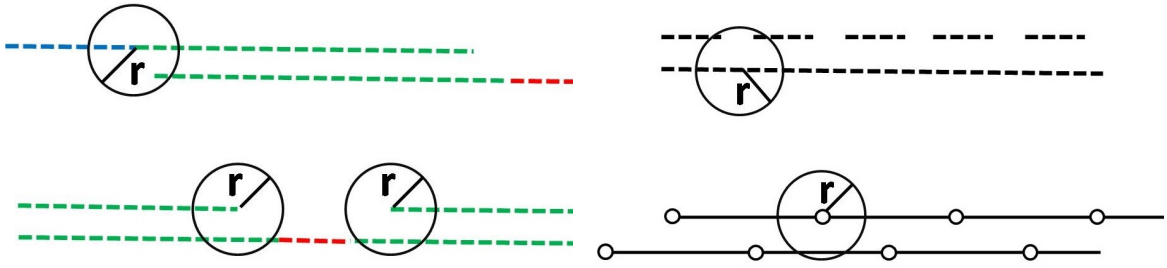


Figure 3.5. (a) Pointwise comparison of two lines. The upper line is considered the reference in this example. Green dots are considered matched with respect to the search radius r. Blue dots are considered false negatives. Red dots are false positives. Note that in the lower example the tracing results in two microtubules instead of one in the reference. The number of false negatives, however, is not significant. (b) Upper example: All points on each line will be considered fully matched for the chosen search radius, although the geometry of the line is not matched at all. Lower example: The sampling of the lines is not dense enough to identify any matches with the chosen search radius.

Search radius for comparison The above definition assumes that line segments that are close to each other model the same filament. Since this depends on the search radius r, we need to estimate a reasonable search radius. We call two points in two different tracings *corresponding* if they are placed by two different tracers in the tomogram to model the same filament. If the two points belong to the lines of two distinctive filaments, we say that they do not correspond. Two points *match* if their distance is below r. The definition of pointwise FNR/FPR requires that two points only match if they also correspond and that points that do not match also not correspond. For a correct computation of FNR and FPR, the distance between two points that correspond must be below the search radius and two points that do not correspond must be further apart. The choice of r must be such that all corresponding points match and no points match that do not correspond. This requires that such a radius exists, which might not always be the case. To find a

[1] Note that FNR and FPR correspond to *precision* and *recall* by $FNR = 1 - \text{recall}$ and $FPR = 1 - \text{precision}$

reasonable search radius, we first determine the minimum distance of two distinctive microtubule centerlines by computing line distances in the tracings. For each point p in a tracing T, we search for a point on a line, other than the line that p belongs to, that matches p with search radius of r. We call the subset of points of T where such a match on a different line is found T_r. We compute the fraction of points in the line set T that have the described *matches*:

$$\text{frac}(r, T) = \frac{\#(\text{points in } T_r)}{\#(\text{points in } T)}$$

We can then plot a curve for each line set by varying r. The minimal distance of distinctive microtubule centerlines can be seen in such a plot were the curves begin to rise from 0 (see Figure 3.7(a)). The search radius must then be chosen below this point. Second, we tried to estimate how large the distance between corresponding points in two tracings is. For each point p in a tracing T_j , we searched for a point in a tracing T_i that matches it. The subset of points of T_j that have such a match in T_i called $T_{j,r}$. We compute the fraction of points in the line set T_j that match points in another line set T_i with search radius r:

$$\text{frac}(r, T_j, T_i) = \frac{\#(\text{points in } T_{j,r})}{\#(\text{points in } T_j)}$$

We can then for each line set T_j plot a curve varying r and T_i. Because we assume that corresponding points are placed at least close to each other, we expect the curves to rise fast for low values of r. At some search radius, all corresponding points also match. We would then expect the curve to rise only slowly when increasing r further, since further matching points that do not correspond are added to $T_{j,r}$. The maximal distance of corresponding lines in different tracings can then be estimated from the curve at the point where its slope drops sharply. This point also gives us the minimal search radius (see Figure 3.7(b)).

Limitation of pointwise FNR/FPR Measuring the amount of matches gives an estimate of the total length of correctly traced line segments, but it is insensitive to line breaks, as depicted in Figure 3.5(a) (bottom example). In the upper example in Figure 3.5(b) FNR and FPR will both be 0 regardless of which line the reference is, even though the upper line is broken into many small pieces. Comparing number of lines is therefore essential for a proper evaluation.

Note that for a correct result, the search radius must be significantly larger than the sampling on the line (see Figure 3.5(b)). We introduce samples as needed.

3.5.2 Generating a ground truth from manually segmented centerlines

We assume that all lines in the manual tracings are true microtubules because we have only rarely observed false positives in the manually traced lines. However, comparing an automatically obtained tracing to a single reference created by an expert bears the danger of overestimating the number of false positives since even experts often miss microtubules completely. To get a reliable quality estimation, we combine several manually obtained results to one set of polygonal lines that approximates best the set of centerlines of microtubules in a volume. In this set, which we will refer to as *ground truth*, each microtubule centerline should be present only once and each traced line in each of the manual segmentations must have a corresponding line in the ground truth.

Picking a representative is difficult, because lines traced by different experts for the same microtubule vary slightly. We mainly observed that lines are of different length and lines are broken. To deal with these inconsistencies and to establish a ground truth, we propose a simple method that consists of two steps: 1) Establish a most likely correspondence of lines in the line sets and 2) compare the length of corresponding lines and choose the longer one.

To establish correspondences of lines in two tracings, lines are sampled densely and matching points are computed. For each line l_o in one line set we find the line l_m of the other set that contains the most matching points. If l_o is longer than l_m we drop l_m, else we drop l_o. The remaining lines form the new set that is in turn compared to the next manual tracing until all manual tracings have been added to the ground truth.

Limits The described method will give inaccurate results when the longer line misses points of the shorter line. Figure 3.6(a) shows this case. The blue and red line correspond and the blue line is longer. However, it misses the end of the red line. Thus, the correspondence of lines might not be established correctly.

The blue line in Figure 3.6(b) intuitively corresponds to the green and the red line. However, above we only compute *one* corresponding line and neglect every other possible occurrence. Suppose the blue and black line are in L1 and green and red are in L2, and blue is processed before black. The following will happen: The red line will be added, because it is longer than the blue line. The black line will be added, because it is longer than the red line. The green line will be neglected, because it has a correspondence (blue line), but was not considered, because the blue line was compared to its most likely correspondence (red line). The result is depicted in Figure 3.6(c).

There are many more possible configurations where the above method would fail to generate a set of lines that one would intuitively expect. To evaluate if we can use the ground truth computed by this method in practice, we show in Section 3.6.1 how much the results computed from our data differ from the expected result. We compute the FNR (Section 3.5.1) for each manual tracing used to generate the ground truth. Ideally, the ground truth would yield an FNR of 0 for each manual tracing.

Figure 3.6. (a)The blue line is longer then the red line and will therefore be added to the ground truth. The right end of the red line will be missing in this ground truth. (b) Configuration that might result in an erroneous ground truth. (c) Resulting line set when computing a ground truth as proposed from the example in Figure (b).

3.5.3 Parameter analysis of the tracing algorithm

Template matching has two parameters (length and mask diameter) and the tracing algorithm has four parameters (C_{min}, S_{min}, d, and w_o) influencing the quality of the obtained tracings. Finding the best settings of these parameters requires first to define a function quantifying the quality of the tracings and then to find a parameter region in the (6-dimensional) parameter space, where this function satisfies a quality constraint for all tomograms to be processed. We use the FNR and FPR measures defined in Section 3.5.1. Since a reference tracing is not available for each tomogram, we find an appropriate parameter region from a subset of tomograms where a reference tracing is available. We call these tomograms test tomograms. Parameter sets yielding the desired quality lie at the intersection of the parameter regions computed for the test tomograms. Note that the quality of a good parameter set depends on how well the chosen test tomograms used for finding the parameters reflect the variation of quality and characteristics of the whole set of tomograms.

Finding the suitable regions to chose the parameters from can be difficult in practice. To obtain good parameters, we therefore first approximate local minima for FPR and FNR in the parameter space for each test tomogram and then set all parameters except for C_{min} to their average. To find the local minima, we vary one parameter and compute FNR and FPR with the other five parameters fixed. We then plot FNR over FPR and from the resulting curve choose the point (FNR, FPR) that is closest to the origin (see Figure 3.9(a)). Since the parameters are not independent, this process needs to be iterated over all parameters, after one parameter has changed, until the process yields no more significant parameter changes. In practice, however, we treat w_o and d as independent from all other parameters and template length and mask diameter as independent from each other. For example, to find the best template length, we only iterate over the three parameters S_{min}, C_{min} and template length. Instead of finding a range for each parameter, we fix all parameters by choosing the average of the best parameter sets found for the test tomograms. For C_{min}, instead of using the average between different tomograms, we examine

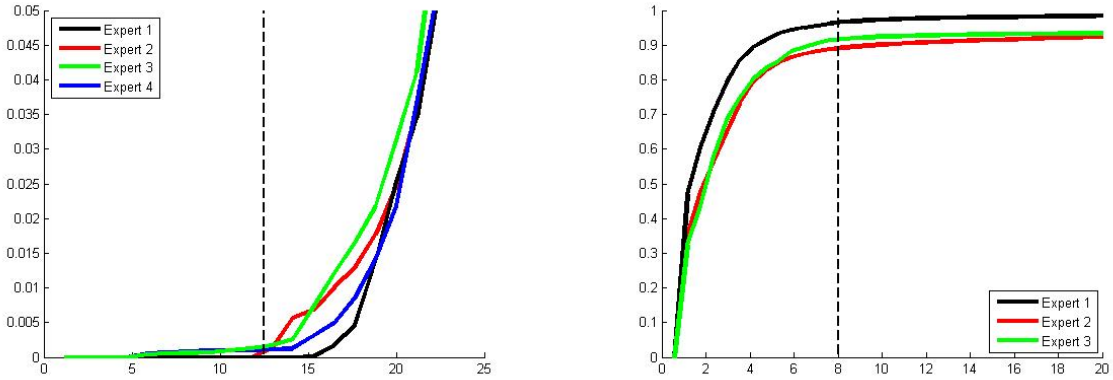
the range of possible values more carefully. We plot FNR and FPR over C_{min} to find ranges of this parameter satisfying the constraints for each test tomogram. We choose the final C_{min} set to lie in the intersection of the ranges for all test tomograms. The reason for the special treatment of C_{min} is that this parameter varies most in the different tomograms, while searching for optima of the other parameters yielded similar results for all test tomograms.

3.6 Results for electron tomograms of plastic-embedded samples

3.6.1 Search radius for comparison, ground truth and inter-user variability

For ground truth generation, we chose two single-axis and three dual-axis tomograms, showing different image characteristics and quality typically observed in electron tomograms. For each of these tomograms, four manually obtained tracings from four different experts were available.

Search radius Figure 3.7(a) shows the plots of $\text{frac}(r, T_i)$ for each manual tracing of centerlines. For a search radius smaller than 12.5 nm, the computed fraction is below 0.005. This means that less than 0.5% of points on microtubules have points closer than 12.5 nm on another microtubule. Accepting an error of 0.5%, we can therefore safely choose r between 0 nm and 12.5 nm. Figure 3.7(b) shows the plots of $\text{frac}(r, T_j, T_i)$ for the tracings obtained by expert 1-3 compared to the tracing of expert 4 (arbitrarily chosen). The slope of the line is large between 1 nm and 8 nm and then quickly decays. We can therefore assume that nearly all lines modeling a microtubule were matched with a search radius of 8 nm. This leaves a safety margin from 8 nm to 12.5 nm from which to choose the search radius.



(a) (b)

Figure 3.7. (a) Plots of $\text{frac}(r, T_i)$ for four manual tracers. (b) Plots of $\text{frac}(r, T_j, T_i)$. T_j is here the tracing of expert 4, which is compared to the other expert's tracings.

Results for ground truth generation We combined the manual tracings to a superset representing all found lines as described in Section 3.5.2. Figure 3.8(a) shows FPR computed with the pointwise comparison (Section 3.5.1) for the four manual segmentations with respect to the generated ground truth. Because a value > 0 for FPR indicates the ground truth was not computed correctly, we visually inspected the 'false positives' on the manual tracings. We found four cases:

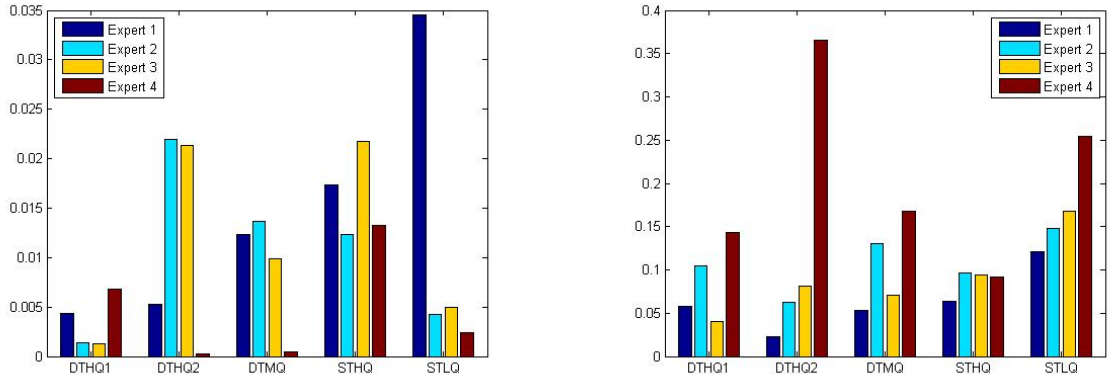
1. The distance between lines in the manually segmentation and ground truth exceeds the search radius r for establishing correspondences of points.
2. The line bends off in the middle of a microtubule and stops.

3. The line bends off at the end of a microtubule.

4. The case illustrated in Figure 3.6(a).

However, a mean FPR of 0.01 (Figure 3.8(a)) indicates that the error is negligible.

To estimate, how many lines are missed by experts, we compared each manual tracing to the computed ground truth. Figure 3.8(b) shows FNR for the manual tracings. Here, mean FNR was 0.12 which means that on average, each expert missed 12% of the sum of line lengths.



(a) (b)

Figure 3.8. (a) Pointwise FPR for manual tracings compared with ground truth. Mean FPR is 0.01. (b) Pointwise FNR for manually tracings compared with ground truth. Mean FNR is 0.12. Search radius was 10nm for both measurements.

3.6.2 Parameter choice

We applied the procedure described in Section 3.5.3 to find minima for FNR and FPR. We use the ground truth for five samples that were generated as described in Section 3.5.2. For computing FNR and FPR, we chose a search radius of 10 nm for matching points, i.e. approximately 50% of the diameter of a microtubule (see Section 3.5.1). The final parameters were: cylinder length = 60 nm, mask diameter = 25 nm, outer cylinder diameter = 20 nm, inner cylinder diameter = 11 nm, $S_{min} = 0.18$, $w_o = 0.3$, and $d = 50$ nm. Figure 3.9(b) shows FNR and Figure 3.9(c) FPR on the five test tomograms with varying C_{min} and the other parameters fixed to these values. From this plot, C_{min} can be chosen from the range 0.27–0.33 for dual-axis tomograms and from the range 0.3–0.33 for single-axis tomograms when constraining $FPR < 0.05$ and $FNR < 0.15$. For all tomograms, we chose the parameter settings mentioned above with $C_{min} = 0.27$ for dual-axis tomograms and $C_{min} = 0.3$ for single-axis tomograms.

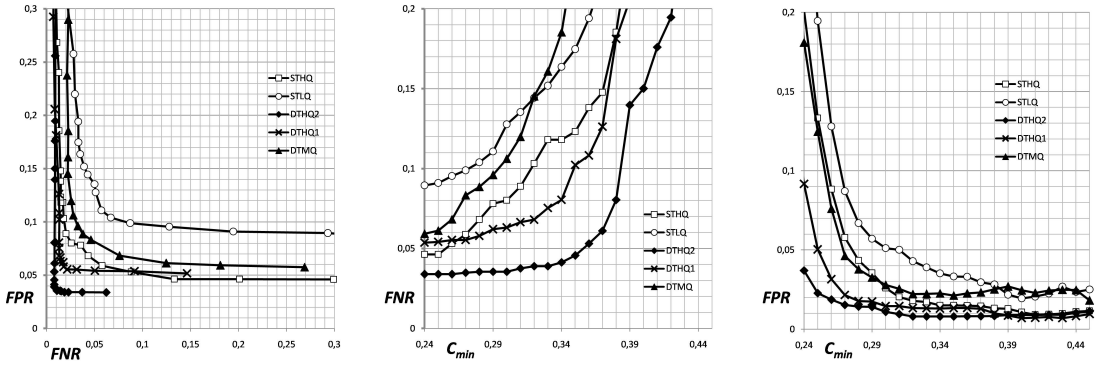
To analyze the stability of the parameter S_{min}, we plotted FNR and FPR against S_{min} in the same manner, fixing C_{min} to 0.27 for dual-axis and 0.3 for single-axis. Our analysis revealed that S_{min} can only be chosen from the range 0.18–0.2 for single-axis from the range 0.18–0.22 for dual-axis tomograms. Results for the five test tomograms obtained using these parameters are shown in Table 3.1. The FPR was below 5% and the FNR was below 7% in dual-axis (below 12% in single-axis) tomograms.

3.6.3 Quality of tracing for *C. elegans* sections

The computation of template matching took between 6 h (Ge-Force GTX 470) and 12 h (GeForce 285) for a tomogram of dimensions 2000 × 2000 × 200. Running the automatic tracing algorithm took 20 min on average. We computed automatic tracings using the parameters from Section 3.6.2 for 26 dual-axis tomograms and 26 single-axis tomograms acquired as described in Section 3.2 with a magnification of 9400×.

	FPR	FNR
DTHQ1	0.02	0.05
DTHQ2	0.02	0.03
DTMQ	0.04	0.07
STHQ	0.03	0.07
STLQ	0.05	0.11

Table 3.1. FPR and FNR of the automatic tracing for the five test data used for parameter tuning when comparing to the reference obtained by combining four manual tracings. Parameters were chosen as described in Section 3.6.2. DTxxx are dual-axis and STxxx single-tilt tomograms. The postfix HQ/MQ/LQ means high/medium/low quality.



(a) (b) (c)

Figure 3.9. (a) FNR plotted against FPR when varying C_{min} from 0.23 to 0.45. (b) FNR plotted against C_{min}. (c) FPR plotted against C_{min}. Results are shown for five tomograms comparing automatic tracing to a reference containing all lines from four manual tracings. Parameters apart from C_min are fixed as described in Section 3.5.3. DTHQ1, DTHQ2, DTMQ: dual-axis tomograms. STLQ, STHQ: single axis tomograms. Single-axis tomograms display lower C_{min} stability than dual-axis tomograms.

We compared the automatic tracings to manually traced line sets that were created by experts using either the editor described in Section 3.4 or IMOD. Here, for each tomogram only one manual tracing was created, which was not verified by another expert. For each automatic tracing, we computed FNR and FPR (Section 3.5.1). Figure 3.10(a) shows FNR plotted against FPR for the dual-axis tomograms. Without removal of outliers, FNR was on average 0.06 and FPR was 0.12.

We closely inspected the results showing $FNR > 0.1$ or $FPR > 0.15$. For the three automatic tracings depicted as triangles in Figure 3.10(a), we observed that most of the lines identified as false positives were actually true microtubules which were missing in the reference. We therefore created a superset of lines containing all microtubules in the automatic and the manual tracings and used this superset as the new reference. To do this, we cleaned up the automatic tracings using the editor described in Section 3.4. We manually checked each line by looking only at the ends, not each point on the line, and removed the line if it was not a microtubule. The removed lines were mostly short lines (<100 nm). We connected lines if they clearly traced the same centerline and occasionally added lines for microtubules that were missed by the automatic tracing (false negatives). For the three tomograms, this took 1.5 h each on average.

Table 3.2 shows FNR and FPR when comparing both manual and automatic tracings to the new reference. The false positive ratio for the automatic tracings (Table 3.2, auto-superset FPR) was on average only 0.06 as opposed to 0.23 before correction of the reference tracings. Thus, the experts missed 17% of the centerlines on average in these three cases (Table 3.2, manual-superset FNR).

	FPR	FNR	FPR(4700×)	FNR(4700×)
Tomogram A (auto)	0.07	0.03	0.05	0.03
Tomogram A (manual)	-	0.22		
Tomogram B (auto)	0.05	0.07	0.07	0.09
Tomogram B (manual)	-	0.15		
Tomogram C (auto)	0.07	0.03	0.07	0.04
Tomogram C (manual)	-	0.14		

Table 3.2. Comparison of manual and automatic tracing to the refined references, obtained as described in Section 3.6.3, for the three results depicted by triangles in Figure 3.10(a). The first two columns show the evaluation for a magnification of 9400×, the last two columns show the result for the automatic tracing for tomograms acquired with a magnification of 4700×.

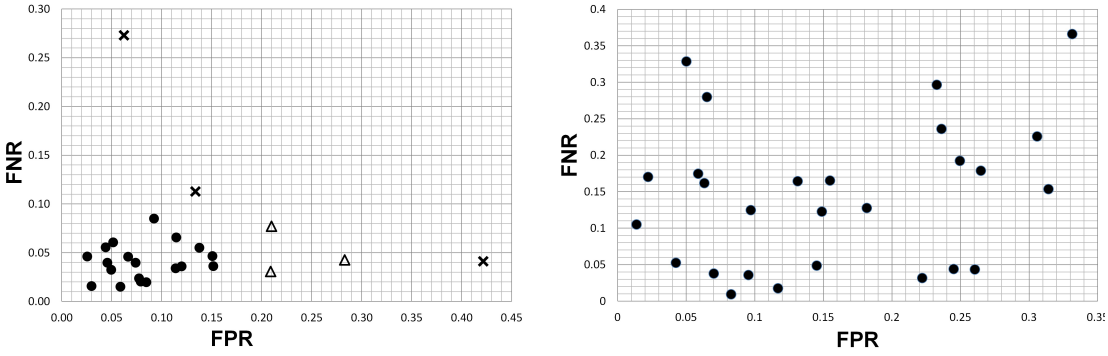
The last two columns in Table 3.2 show FNR and FPR for the same three samples comparing the automatic tracing with the described parameters to the refined reference for tomograms acquired at only half the magnification (4700×) and therefore half the resolution. On average FPR was 0.06 and FNR was 0.05. This shows that the microtubule tracing algorithm remains remarkably reliable at low resolution (2.5 nm pixel size). In contrast to the three results depicted as triangles in Figure 3.10(a), the three crosses represent tracings of tomograms where the high FNR and/or FPR values could not be explained by an erroneous reference. Figure 3.11 shows one slice through these tomograms. In comparison to a typical tomogram (Figure 3.11), the quality of these three tomograms (Figure 3.11(b)-(d)) is clearly lower, showing blurriness (3.11(b)), strong cluttering (3.11(c)) or ice crystals (3.11(d)). This suggests that the automatic tracing is more sensitive than a human expert to low quality data. These tomograms (3 out of 26) could easily be excluded from the analysis just by visual inspection. After removing the three outliers and with using corrected references for tomograms in Table 2 (also triangles in Figure 3.10(a)), the average FPR of 23 tomograms was 0.08 and the FNR was 0.04.

The FPR and FNR values account for the accuracy of the total length on microtubule tracings, not for the accuracy in the number of microtubules (see Figure 3.5(a)). Thus, we also checked how reliable the number of automatically traced microtubules was. We computed the ratio of the number of lines in the automatically traced microtubules to the number of lines in the references for all tomograms, except for the outliers depicted by crosses in Figure 3.10(a) using the newly generated references for the three outliers depicted as triangles in Figure 3.10(a). On average, the ratio was 1.24 ± 0.03 (mean ± SEM, $n = 23$), which means that the automatic tracing overestimated the total number of microtubules by 24% compared to manual tracings.

In order to test the automatic segmentation on single-axis tomograms, we applied the same comparison to 26 single-axis tomograms. As discussed before (see Section 3.6.2), we set $C_{min} = 0.3$. Figure 3.10(b) shows FPR plotted against FNR for all 26 test data. The average FPR was 0.17. Again, for a few randomly picked examples, most of the false positives could be identified as true positives, but this was not quantified. The average FNR was 0.16, which is 2–3-fold higher than in dual-axis tomograms. As expected, these results suggest that single-axis tomograms are much less suitable for automatic tracing of microtubules than dual-axis tomograms.

To understand the influence of the tilt axis orientation on tomogram quality, we analyzed the angular dependency of FNR and FPR in the x–y-plane (see Figure 3.12). For dual-axis tomograms, FNR and FPR were independent of the orientation. For single-axis tomograms, both measures increased substantially for orientations perpendicular to the tilt axis.

The tomogram quality could potentially differ at top and bottom of the tomogram as a result of the defocus, although this effect is not as significant for electron tomography as it is, for example, for confocal microscopy. Thus, we measured FNR and FPR at the top and bottom 100 nm of the tomograms separately. The largest difference we found was only 0.007 in dual-axis tomograms when measuring FNR. The difference in FPR was even lower.



(a) (b)

Figure 3.10. (a) FNR/FPR for 26 dual-axis tomograms. Triangles: manual tracings were incomplete. Crosses: Volumes showed preparation or imaging artifacts (see Figure 3.11). (b) FNR/FPR for 26 single-axis tomograms.

3.6.4 Quality of tracing for other samples

To get an idea of problems that might arise when varying specimens and protocols, we tested the extraction algorithm on (a) Drosophila epithelial cells, (b) chemically fixed HeLa cell mitotic centrosomes, and (c) HeLa cell midbodies, acquired on the same microscope with a different camera. No manually generated references were available for these tomograms. Here, we give a qualitative description of the results when running the pipeline as described with the same parameters as for *C. elegans* unless stated otherwise.

a) In contrast to the *C. elegans* samples, microtubules in the Drosophila cell sections vary in shape and the tomograms contain membranes. Since the template matching assumes a constant diameter of the microtubules (we measured 22 nm as opposed to 20 nm in *C. elegans*) throughout the tomogram, as expected, microtubule regions deviating from the assumed diameter were missing from the tracings. We assume this shape variation of microtubules might have been caused by the high pressure freezing (see Section 2.2.3). The computed correlation at membranes was sometimes high, in particular if two membranes were separated by 20 nm. This caused false positives (see Figure 3.13 and 3.14).

b) For the chemically fixed HeLa cells, the algorithm seemed to perform equally well as for the high-pressure frozen cells with respect to false positives and false negatives. However, chemical fixation causes strong bends in microtubules that resulted in breaks in the tracing, because the model assumes microtubules are relatively straight. Dropping the linearity parameter from the tracing and opening the angle of the search cone to 100° compensated this effect partly, but sharp kinks still caused breaks. Then, some microtubules running close to each other, for example, the centriole microtubules, were connected by a bending line (see Figure 3.15).

c) The HeLa midbody samples show the midbody and bundles of microtubules still attached to it (see Figure 3.16(a)). Contrast and resolution was lower in these samples (see Figure 3.16(c)). In the midbody, microtubules are embedded in electron dense material of yet unknown origin. Microtubules were detected outside the midbody by the automated method but not inside the midbody (see Figure 3.16(b)). Again, some false positives are present where the sample contains membranes. Some microtubules could not be detected by the method, though we cannot quantify the amount of false negatives. We found that a larger radius for the template (24 nm) yielded better results than the 20 nm used for the chemically fixed HeLa cells. The differences in contrast and apparent microtubule diameter might result from different cameras and calibrations between different microscopes.

3.7 Technical note

The automated segmentation approach described here has been integrated in the visualization and data analysis software Amira (Stalling et al. [127]). Compiled binary versions (for Windows and Linux) of the microtubule segmentation package are available for use in research via the Zuse Institute Berlin upon electronic registration under
http://www.zib.de/en/visual/software/MicrotubuleSeg.html.

The five test tomograms from Section 3.6.2 and the generated references are available at
http://publications.mpi-cbg.de/publications-sites/4629/ElectronTomograms.html.

3.8 Discussion

The automatic technique for microtubule centerline extraction that we described allows a much quicker segmentation of microtubules in a large number of samples than manual segmentation. Our results suggest that the total amount of microtubules can be measured without manual interaction from dual-axis tomograms with ~4% false negatives and ~8% false positives on average. For tomograms of sufficient quality, the upper bound for FNR/FPR is 0.1/0.15 (Figure 3.10(a)). The analysis of some samples revealed that false positives actually include true microtubules that were missed by manual segmentation (see analysis of outliers depicted by triangles in Figure 3.10(a) and Table 3.2). This suggests that the automatic tracing might be more reliable than individual human experts, who missed up to 17% of microtubules in the worst case. However, we also observed that the automatic tracing makes certain types of errors. For example, it creates broken lines that an expert can easily correct. The automatic tracing errors become higher when the tomogram quality decreases (outliers depicted by crosses in Figure 3.10(a)). Thus, a manual verification is always advisable.

For the majority of tomograms, both false negatives and false positives were below 10%. We believe that this quality is sufficient to address biological questions on the density of microtubules, for instance in comparing different biological conditions, such as wild type and mutant or RNAi samples. If accurate number and lengths of microtubules in a specimen are required, a manual correction of the automatically traced microtubules seems necessary. The automatic microtubule reconstructions often contain breaks and short false positives. This results in an overestimation of the number of microtubules and an underestimation of their lengths. This effect might be even more severe when the microtubule's shape deviates from the shape of a cylinder, an effect caused, for example, by katanin (Srayko et al. [6]). For the three examples that we checked, we observed a significant number of short false positives and on average, the automatic tracing overestimated the number of microtubules by 24%. However, the three manual segmentations used as references were obtained by single experts and only 3 of the 23 tomograms (triangles in Figure 3.10(a)) were corrected. Thus, the reference tracings probably have false negatives and the effective overestimation in the number of microtubules is likely to be less than 24%. Fortunately, an expert can correct breaks and false positives in a reasonable amount of time (approximately 1.5 h in our experiments) using the editor that was specifically designed for this task (Section 3.4).

We believe that 1.5 h is the time needed for microtubule centerline segmentation with the proposed method by a trained expert on a high quality tomogram not showing severe deformations or artifacts. This amount of time is reasonable compared to the time required for specimen preparation, image acquisition and tomogram reconstruction as carried out today (several hours per tomogram). It is also a major improvement to the fully manual segmentation currently used. Thus, the manual verification step is not the bottleneck in the workflow of 3D reconstruction of microtubules by electron tomography. As determined from three manually segmented tomograms (see Figure 3.8 and Table 3.2), the false negative rate in manual segmentation varies greatly and can be relatively high (up to 17% here, on average 12% for the test data). This variability is expected from human decisions, and it is likely to affect the quality of any large-scale study. In the two-step approach, involving automated tracing of microtubules followed by manual correction, the final result is, we believe, more accurate than either the manual or the automated segmentation alone. Indeed, the software will produce more consistent results than humans, requiring only a small number of manual corrections.

We measured a substantially larger FNR – about 3-fold – in single-axis tomograms compared to dual-axis tomograms (average FNR: 0.15 in single-axis vs. 0.04 in dual-axis, see also Figures 3.10(a) and 3.10(b). These results clearly show that without any further heuristics, the described method cannot be used for quantitative analysis of microtubules in single-axis tomography. A more severe problem is that the visibility of microtubules depends on the orientation of the microtubules relative to the tilt axis (see Figure 3.12). Microtubules oriented perpendicular to the tilt axis will not be visible in the tomogram due to the missing wedge since the signal is lost nearly completely. Both manual and automatic analysis will be unreliable in this case. Even if we had achieved a lower FNR, the automatic tracing might still miss many microtubules that human analysis missed too. Thus, the limitations of single-axis tomography make it difficult to estimate the true error of the automatic tracing. Since the distribution of orientations, in general, is not known a priori, the number of microtubules extracted from single-axis tomography can, in general, only be an unreliable underestimate of the true number. We therefore recommend using dual-axis tomography for collecting numbers and distribution of microtubules. This has been recognized in manual segmentation already (Mastronarde [87]), and it is even more the case when using our automated tracing approach. Only if an a priori assumption can be made about the spatial distribution of microtubules might single-axis tomography be an option. It is faster (reduced acquisition and reconstruction times), but the data analysis requires more care.

Importantly, all results depend on the parameter choice for the template and the tracing. The stability analysis revealed that C_{min} and S_{min} have to be chosen within narrow margins. The margin for choosing C_{min}, for example, is 0.02 for single-axis acquisition and 0.05 for dual-axis acquisition of the five test tomograms (see Section 3.6.2). Because every change in the protocol and image acquisition parameters influencing image characteristics might result in a different correlation field, parameters might have to be adjusted again when changing the acquisition. Also, to ensure a parameter set can be chosen at all, sample quality must be sufficiently consistent.

The algorithm could also be used to find other line-like structures, such as intermediate filaments, but in this case it is even more crucial to study the best parameter setting thoroughly. The method described in Section 3.5.3 for the determination of optimal parameters is tedious and time-consuming. Preliminary results show that the cumbersome process of tuning parameters might in the future be replaced by interactive tools (Torsney-Weir et al. [128]). Furthermore, parameter tuning requires that reference tracings for several tomograms are available. Thus, the automated approach with fully tuned parameters is time efficient only when a large enough number of tomograms, acquired in similar conditions, need to be analyzed. A good application for this is a large number of samples, for statistical analysis or correlation studies.

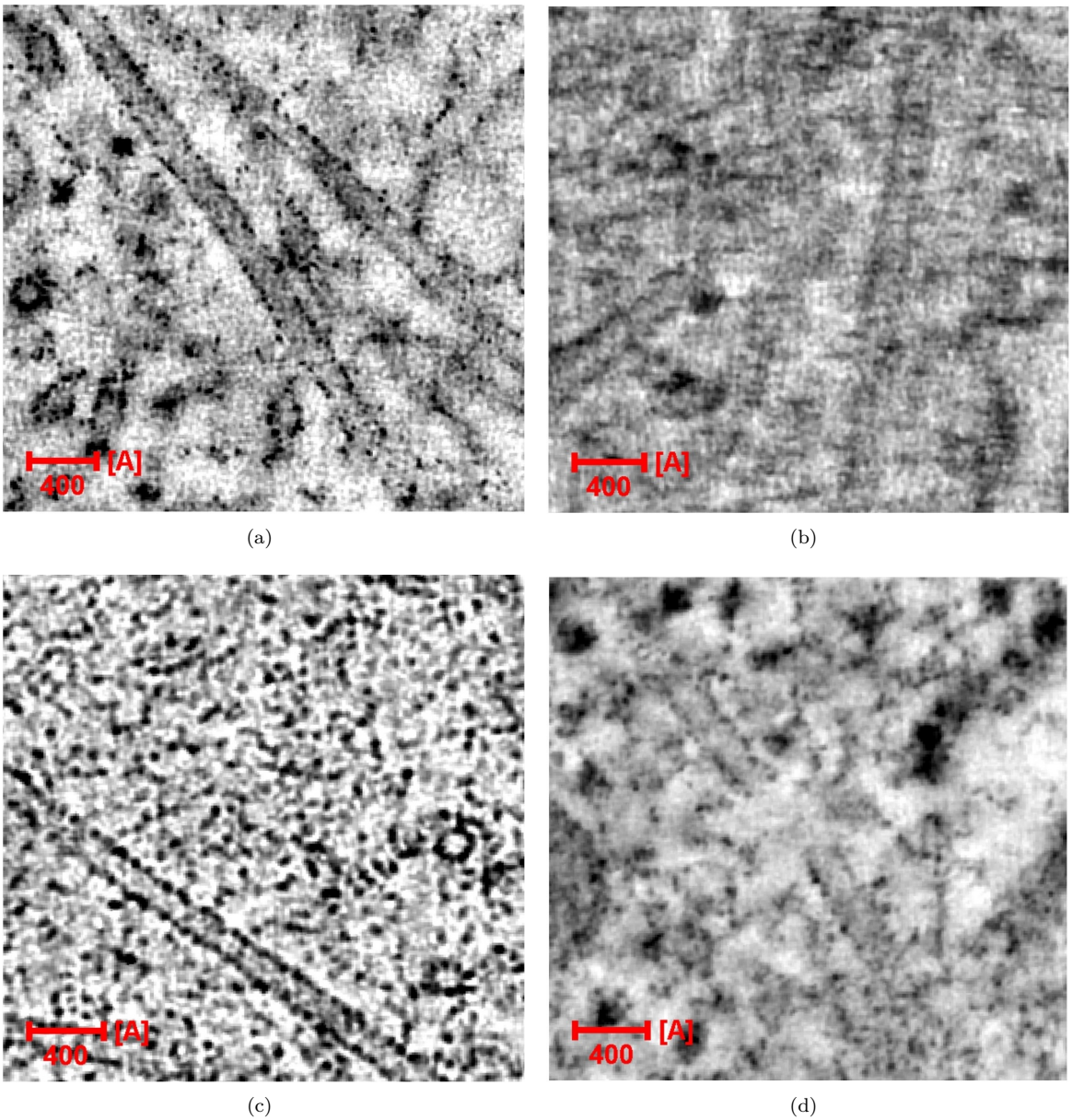
Altogether, our approach is based on the assumption that microtubules are rigid cylinders. In some samples that we investigated, microtubules were not always rigid but strongly bent along the cylinder axis (chemically fixed HeLa cells), or they were radially deformed (Drosophila epithelial cells). In regions where large deviations from the model occur, tracing might fail. Conversely, objects that locally resemble rigid cylinders are sometimes detected as microtubules. We have seen such examples with membranes. All of these errors can easily be corrected manually. In the midbody samples, the tracing failed inside the unknown electron dense material of the midbody. We believe this is because the model assumption that microtubules appear as dark microtubules on light background is violated in these tomograms.

3.9 Conclusion

The automated tracing of microtubules presented here was successfully applied to achieve a complete and reliable segmentation of dozens of dual-axis tomograms in a reasonable amount of time. When necessary, fully automated tracing can be completed by manual correction using a specially designed editor. Even if manual correction is required, the described approach yields a significant speed improvement over purely manual segmentation of microtubules in electron tomograms. Because of the missing wedge, the automated tracing is less suited and not recommended for single-axis electron tomograms.

For the segmentation of structures larger than one tomogram, several tomograms need to be joined. Microtubules in the z-direction of a stack will in most cases extend over several tomograms.

In the next chapter we describe how we approach this problem.



(a) (b)

(c) (d)

Figure 3.11. Analysis of tomogram quality in dual-axis outliers. (a) Tomogram of good quality ($FNR = 0.05$, $FPR = 0.03$). (b) Tomogram shows blur causing high FNR ($FNR = 0.24$, $FPR = 0.18$). (c) Microtubules are clearly visible in this tomogram, but noise causes many false positives ($FNR = 0.05$, $FPR = 0.41$). (d) Tomogram contains breaks, possibly due to ice-crystals, causing a high FNR ($FNR = 0.11$ $FPR = 0.13$). (b), (c) and (d) are slices through the three tomograms that are depicted as crosses in Figure 3.10(a). None of the samples was digitally filtered.

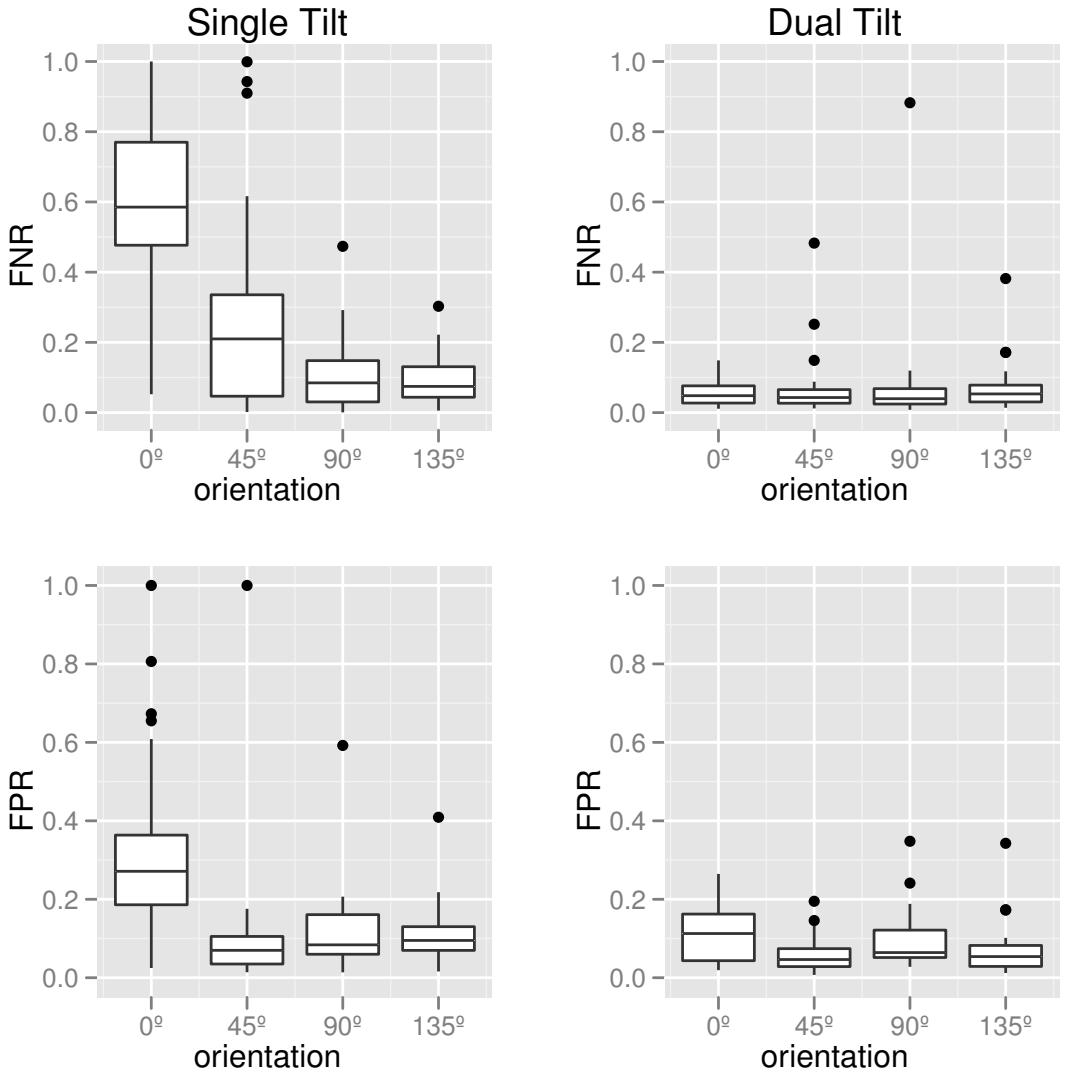
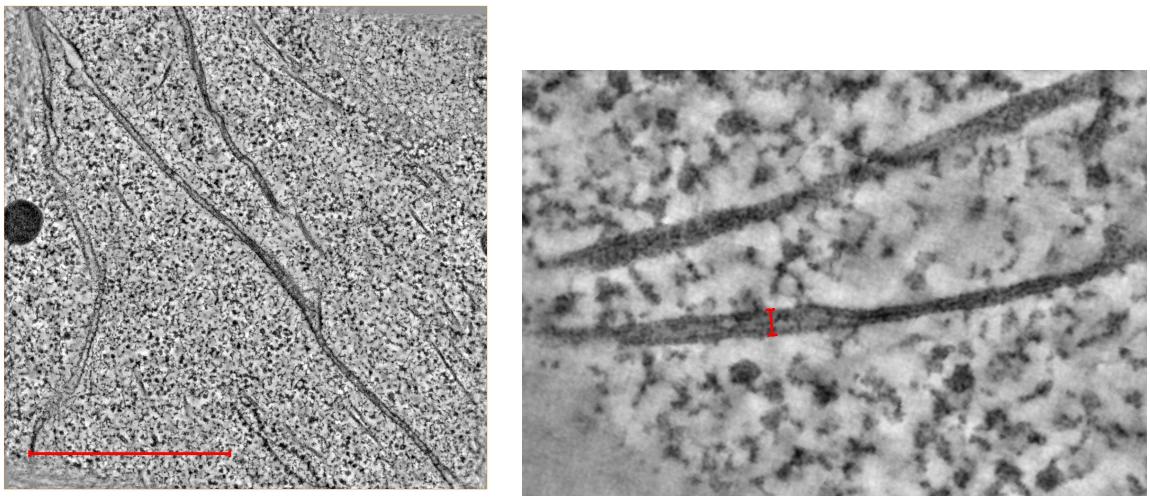


Figure 3.12. Box and whisker plot of angular dependency of FNR and FPR for 27 single tilt and 26 dual tilt tomograms. The whiskers indicate the lowest and highest datum within the 1.5 interquartile range from the lower and upper quartile. FNR and FPR were separately computed for each tomogram for four orientations in the x–y-plane (0°, 45°, 90°, 135°; 0° is perpendicular to the tilt axis; 90° is parallel to the tilt axis). For each orientation, line segments only contributed when the angle between their tangent and the orientation vector was below 22.5°.

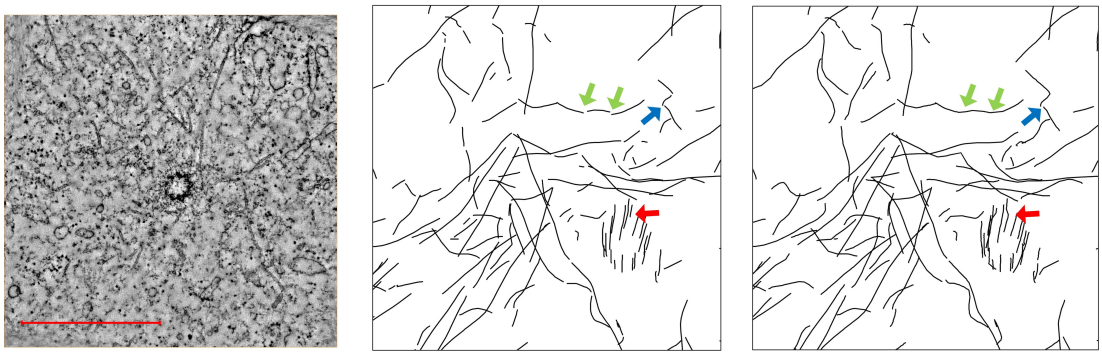


(a) (b)

Figure 3.13. (a) One slice through one of the Drosophila epithelial cell sections. The scale bar is 1 μm. (b) Slice through a deformed microtubule in Drosophila epithelial cells. The scale bar is 22 nm.

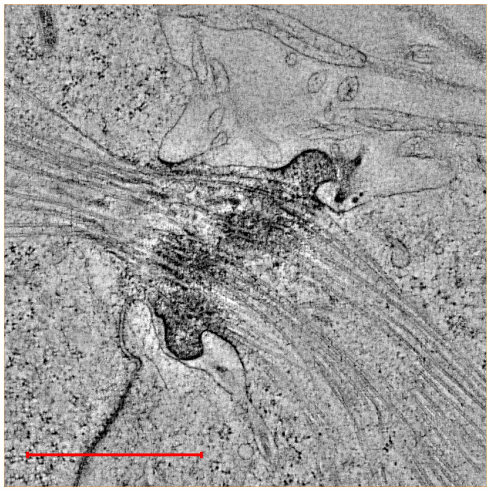
(a) (b) (c)

Figure 3.14. Example for false positives induced by a membrane. In this example, two membranes run at a distance of 20 nm which causes a high correlation with a cylinder of 22 nm in a wide area. (a) Slice through the membrane. The red scale bar is 20 nm. (b) Isosurface of correlation values at threshold 0.27. Since the seed points for the tracing are chosen at this threshold, each voxel enclosed in this surface is a potential starting point for the tracing. (c) Traced lines.

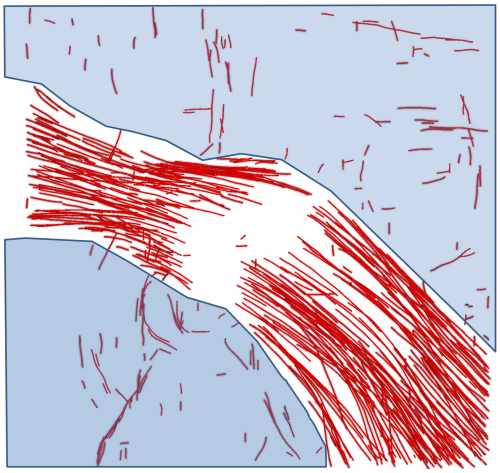


(a) (b) (c)

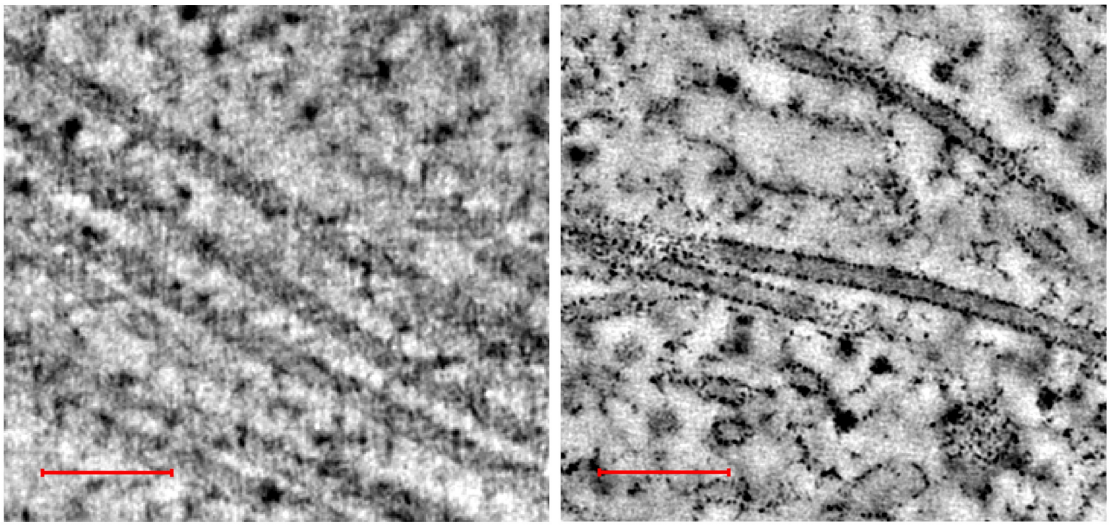
Figure 3.15. (a) Slice through one section of the chemically fixed HeLa cells. The scale bar is $1\,\mu m$. (b),(c) Automatically obtained tracings with and without linearity parameter. (b) Automatically traced centerlines obtained with standard parameters. (c) Tracing obtained when dropping the linearity parameter. Arrows indicate: green) breaks that were correctly closed when dropping the linearity parameter, blue) breaks that were not closed, red) two microtubules are connected falsely – the tracing jumps from one centerline to the other when dropping the linearity parameters.



(a)



(b)



(c)

Figure 3.16. (a) Slice through one section of a HeLa midbody sample. The scale bar is 1 μm. (b) Result of the automated tracing for a HeLa midbody sample. Microtubules outside the marked area are mostly false positives caused by membranes. In the midbody, hardly any microtubules are detected. (c) Left: Slice through HeLa midbody sample outside the midbody. Right: Example slice through a tomogram of the *C. elegans* centrosome samples. The HeLa sample has lower contrast and seems blurred; microtubules are hardly recognizable as tubes in some areas. The tracing misses some of these microtubules. The scale bar is 100 nm.

Chapter 4

Stitching tomogram sections

In the last section, we presented a method to segment microtubule centerlines from electron tomograms. However, tomograms are physical sections of a sample and the previously described method results in a stack of line sets each of which only contains a portion of the microtubules as illustrated in Figure 4.1(a). In order to obtain detailed information on the microtubule organization, such as length and number, the line sets have to be aligned and the ends need to be stitched to obtain a complete model as shown in Figure 4.1(b).

We can choose between two options to obtain a full 3D reconstruction of all microtubule centerlines:

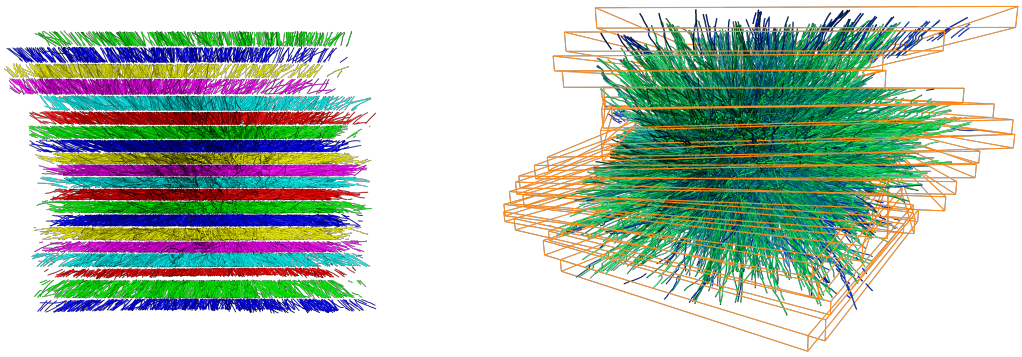
1) Use the top and bottom slices of subsequent tomograms for registering the volumes based on the pixel values of these slices and segment the microtubule centerlines from the stitched tomograms afterwards; or

2) First segment the microtubule centerlines in the individual tomograms and then stitch the line ends.

Because the stitching problem arises in many applications in computer vision and image processing, both approaches are subject to intensive research.

As for 1), Modersitzki points out [129] that a registration of two images requires finding a *reasonable transformation* such that the transformed image is *similar* to the reference image. While it is already challenging to infer a reasonable transformation model for the top and bottom slices of two tomograms, finding a proper notion of *similarity* for electron tomograms is even harder, as we will discuss now. A common way of measuring similarity is to define a pixel value based distance of two images. However, this requires that subsequent sections have similar pixel values. This might not be the case for tomograms: To begin with, the sample deforms while imaging (see Section 2.2.3). Therefore, top and bottom slices of two consecutive tomograms do not necessarily show the top and bottom of the sample but instead a slice through the bent sample. In addition, it has been observed that while sectioning samples and imaging, material is lost (see Section 2.2.3). Prominent biological structures in electron tomograms such as vesicles of membranes are therefore unreliable features for registration as illustrated in Figure 4.2(a) and 4.2(b). Furthermore, the sample might not contain enough features to perform a registration with the required accuracy. For example, the tomograms taken from *X. laevis* spindles contain very few distinct features, as shown in Figure 4.2(d) and 4.2(e). Organelles are missing here, because the egg extract the spindle grows in is purified before the spindle builds (see Section 2.3). Finally, dirt on the surface of a sample, gold particles or contrast variation, all of which are common in electron tomography, can introduce even more errors when computing a distance measure based on pixel values (see Section 2.2.3).

Segmented microtubule centerlines, in contrast to the image data, make excellent features for stitching. They are distributed across the data nicely, since the microtubules are the center of attention when imaging. Usually they are also numerous. Image artifacts like contrast irregularities, dirt, etc. cannot influence the distance measure any further. But, most important, if we use the microtubule centerlines, we can also exploit information about their orientation to help stitch the data together. Figure 4.3 illustrates this with two examples.



(a) (b)

Figure 4.1. (a) Sections of microtubule centerlines in the mitotic spindle of a *C. elegans* early embryo. The centerlines were segmented in electron tomograms. The sections are not aligned and the ends are not connected. (b) The resulting aligned and stitched microtubules. Orange boxes indicate the rotation and shift of the original position of the sections.

To the best of our knowledge, the only software approach to stitch electron tomograms is a semi-automatic tool integrated to the IMOD software (Kremer and Mastronarde [96]). Here, tomograms are transformed using manually selected line ends as landmarks for a thin plate spline warp. Microtubules are traced afterwards in the stitched tomograms. This approach has been used successfully, for example, by McIntosh et al. [130] and O'Toole et al. [131], [7]. However, selecting matching microtubules manually is very time consuming, tedious and, depending on the contrast of the tomogram and the availability of features that could guide the user to potential matches, it can even be infeasible. In addition, the size of a stitched tomogram is likely to exceed the main memory available on a typical workstation since individual tomograms can have a size up to 15GB and, depending on the size of the sample, between 5 and 50 sections are needed to cover the full sample.

We developed an application that stitches microtubule centerlines as opposed to stitching tomograms. While the problem of using coordinates of points for stitching has been studied exhaustively (see Section 4.3), to the best of our knowledge the usage of additional information on orientation for stitching has received little attention. We exploit this additional information in our stitching application. The method consists of three steps:

1) Compute an initial coarse alignment of the ends (alignment problem). For the initial alignment, we identify cliques in distance compatibility graphs, a common technique used in molecular similarity analysis that has also been applied in other applications for aligning points sets (for example, Dercksen et al. [132]).

2) Calculate a refined transformation of line ends to correct for shift and deformation of adjacent sections (registration problem). To find a proper registration of the endpoints, we make use of the coherent point drift (CPD) algorithm, a probabilistic variant of the iterative closest points algorithm, that is described in Myronenko and Song [17]. We show a simple way to integrate the line orientation into the statistical framework underlying the CPD.

3) Find the biological correct connections of the ends (matching problem). We compute the final matching of endpoints using a Markov random field as described in Amat et al. [86]. The correspondences are computed with belief propagation on a factor graph. Because belief propagation does not necessarily converge to a minimum on our test data, we propose to detect the critical nodes in the graph and show that minimal, although carefully selected, user input is sufficient to achieve convergence for our application.

Figure 4.4 shows the three steps of our pipeline. In summary, we make the following contributions:

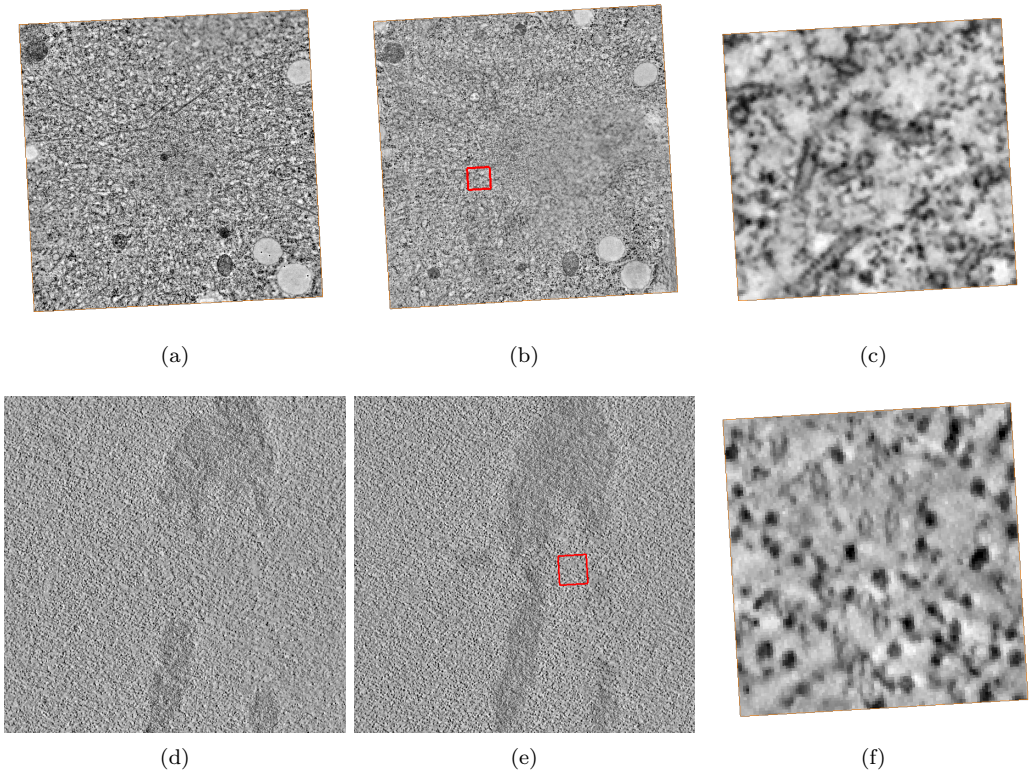


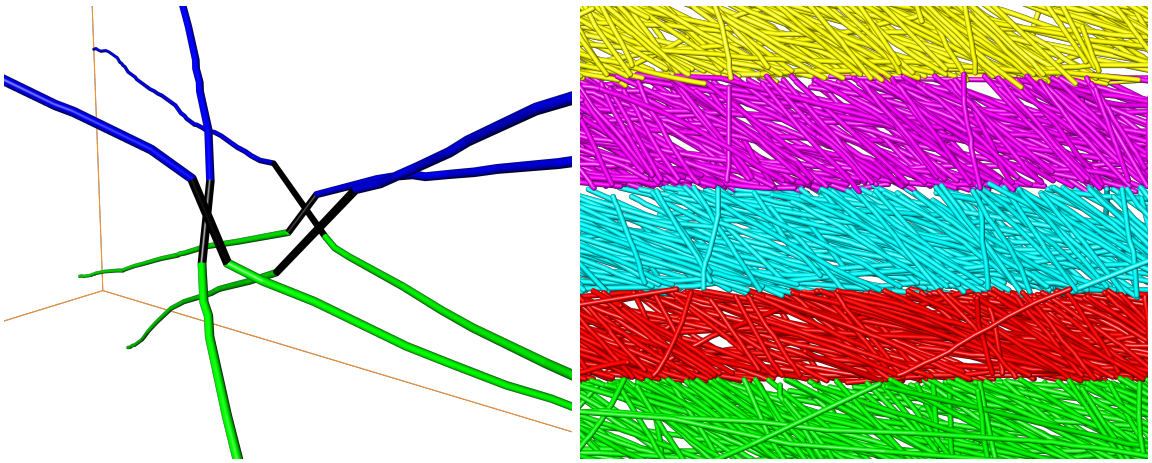
Figure 4.2. (a) and (b) show top and bottom slices of two consecutive tomograms of a *C. elegans* sample. Note that some of the cross sections of the vesicles are not present in both slices. (d) and (e) show parts of the top and bottom slices of two consecutive tomograms of an *X. laevis* sample. The gray patches are the chromosomes. The black dots are mainly ribosomes. None of these features can serve for reliably measuring image distance. (c) and (f) show close ups of the images marked by red boxes in (b) and (e), respectively.

- We describe a pipeline that allows stitching of microtubule centerlines with less that 5% wrong or missing connections.
- We show how orientation of lines can boost performance when employing it in a probabilistic approach for registration (Section 4.4.2).
- We describe a way to detect critical regions in a factor graph when running belief propagation and show, that little user input is sufficient to enforce convergence when solving the matching problem for this application (Section 4.4.3).

We begin this chapter with a summary of the major challenges faced when stitching tomograms and microtubule centerlines. In Section 4.3, we first give an overview on previous work and then elaborate on details of the methods our application builds on. We then show how these methods can be adapted for the stitching of line ends in Section 4.4.

Finally, we evaluate the validity of our model assumptions by comparing automatically obtained results to manually obtained results on true biological data in Section 4.6. We will use electron tomography data of microtubules in three biological samples: microtubules from *Caenorhabditis elegans* mitotic spindles in early embryos (*C. elegans*), microtubules within *Xenopus laevis* meiotic spindles (*X. laevis*) and the sub-pellicular microtubule skeleton of *Trypanosoma brucei* (*T. brucei*).

Large parts of this chapter, such as parts of the previous introduction, are taken from our publication '*Automated stitching of microtubule networks in serial sections of electron tomograms.*' [100]. This paper is in preparation and has not been submitted for publication by the time this thesis was submitted. The copied passages are not explicitly annotated.



(a) (b)

Figure 4.3. Examples for the informativeness of line orientation for finding correspondences. Line color indicates different sample sections. (a) Microtubule centerlines in a centrosome of a *C. elegans* early embryo. (b) Microtubule centerlines in several sections taken from a spindle of *X. laevis*. In both cases the line orientation helps judging the quality of the alignment and finding end correspondences.

4.1 Challenges

We have seen in Section 2.2.3 that tomograms can be distorted in several ways: Samples might be compressed during microtome sectioning and the sample sections warp and twist while imaging as a consequence of the long electron beam exposure (see Figure 2.8). Tiled imaging of large samples might introduce further distortions (see Figure 2.9). Any method used for registering the data must be capable of handling non-linear deformation.

A model for non-linear deformation might reduce the error but biological data is complex and finding a transformation model that captures all effects in all data is rarely feasible. Therefore, end pairs in subsequent sections can still be locally shifted after registration. From previous experiments (Section 3.6) we also know, that the segmentation can't recover the position of a centerline end perfectly. Unfortunately, microtubule centerlines can have a small distance of only 40 nm (Lacomble et al. [133], Höög et al. [5], [134], McDonald et al. [135], Ding et al. [8]) and often run in bundles in which case it is even hard for an expert to decide on the correct pairs (Figure 4.5(c) and 4.5(f)). This makes it difficult to select the correct connections of line ends. The problem is illustrated in Figure 4.5(a) and 4.5(d). Furthermore, we have seen that the segmentation might miss microtubules. This also complicates the decision, as shown in Figure 4.5(b) and 4.5(e). To compute a correct matching, an algorithm must be resilient to local shifts as well as missing data.

The needed accuracy of the stitching depends on the information that has to be retrieved from the final microtubule centerline set. In the worst case, for measurements of the number of microtubules, the errors rate for two subsequent sections add up for the full stack if all lines traverse each tomogram.

However, modeling the problem is not the only challenge when choosing a suitable method for solving the stitching problem. The ultimate goal of developing a stitching method is providing a software that enables biologists to analyze data efficiently with a high accuracy. This imposes further constraints on any algorithm used.

First of all, the application should be designed to be usable by experts in biology. Any parameter setting needed by an algorithm should therefore reflect known properties of the data that can be intuitively understood by an expert. For example, a distance threshold is easy to understand, whereas a regularization weighting parameter in an optimization framework is unpredictable for a user unfamiliar with the optimization framework.

A further requirement for usability is the time needed by a user to stitch the sections. The pure

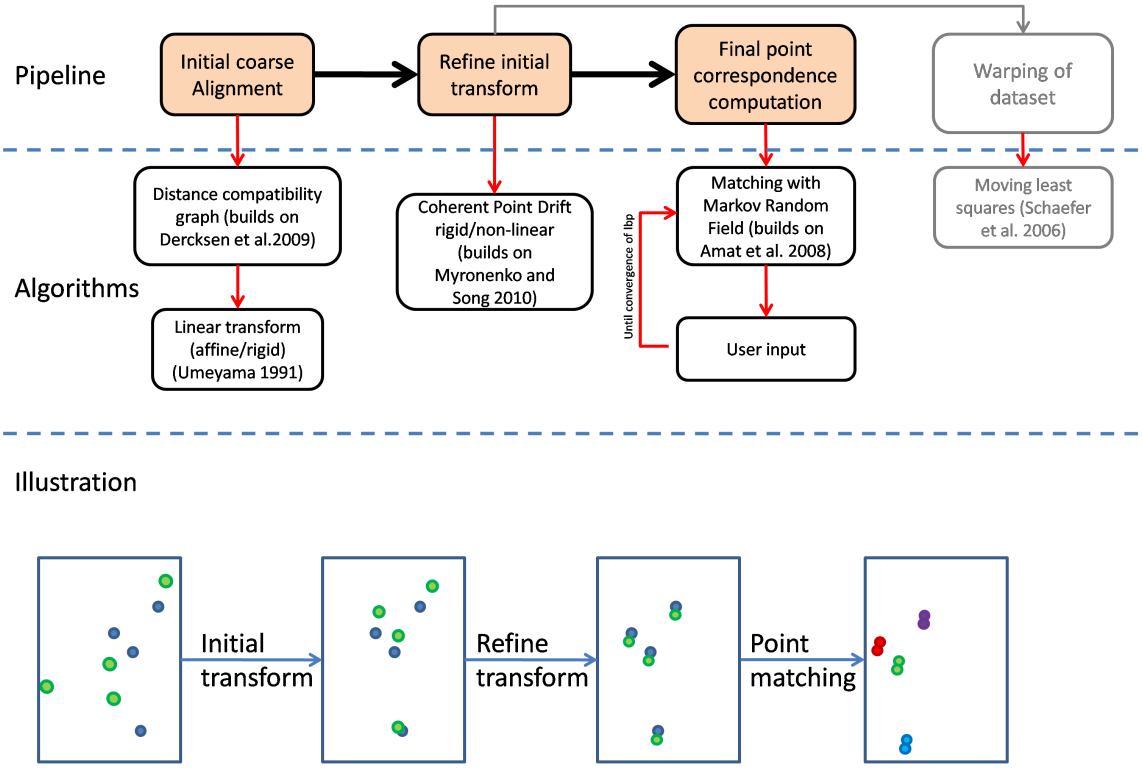


Figure 4.4. The workflow for stitching stacks of microtubule centerlines. The top row shows the four steps. The second row names the implemented algorithms. Citations in brackets give the work the developed algorithms build on (prefix 'builds on') or just names the used methods if we present no modification. The bottom row illustrates the tasks solved by the presented methods. The step 'warping of dataset' is an optional step that can be implemented if a registration of the domain is needed after endpoint registration.

CPU time is not critical in this application because the tomogram generation is a slow process in general. Waiting for a result even for a few days would thus not slow down an expert. However, the effective time spent interacting with the application should be minimized. Here, the challenge is to find the best trade off between accuracy and required user interaction.

4.2 Notation and abbreviations

We refer to the set of line end coordinates $\{x_1, ...x_N\}$ and $\{y_1, ...y_M\}, x, y \in \mathcal{R}^2$ as X and Y. Here, we assume that all line ends lie on a plane and therefore the coordinates of the line ends are two-dimensional. The corresponding sets of orientations $\{\vec{x}_1, ..., \vec{x}_N\}, \{\vec{y}_1, ..., \vec{y}_M\}, \vec{x}, \vec{y} \in \mathcal{R}^3, ||\vec{x}|| = ||\vec{y}|| = 1$ will be denoted as $\vec{X}, \vec{Y}$. We assume that X and $\vec{X}$ contain N and Y and $\vec{Y}$ contain M elements. The correspondence between a line end and the line's orientation is given by the index, that is, the corresponding orientation to x_i is $\vec{x}_i$. For two sets of points $x_i, ..., x_N$ and $y_i, ..., y_M$ the matrix $\mathbf{X}$ contains all points in X as rows $\mathbf{X} = (x_1, .., x_N)^\top$ and likewise $\mathbf{Y} = (y_1, .., y_M)^\top$, $\vec{\mathbf{X}} = (\vec{x}_1, .., \vec{x}_M)^\top$ and $\vec{\mathbf{Y}} = (\vec{y}_1, .., \vec{y}_M)^\top$.

By $\mathcal{T}(y, \Theta)$ we denote a transformation with parameters Θ. For example, if $\mathcal{T}$ is a similarity transform $\mathcal{T}(y, \Theta) = sRy + t$ with scaling s, a rotation matrix R and a translation t, then Θ is the set $\{s, R, t\}$.

For the registration, we introduce a continuous random variable $\mathbf{x}$ and a discrete random variable $\mathbf{m}$ that has states $\{y_1, ..., y_M, M + 1\}$. We write $P(\mathbf{x})$ to refer to the probability density

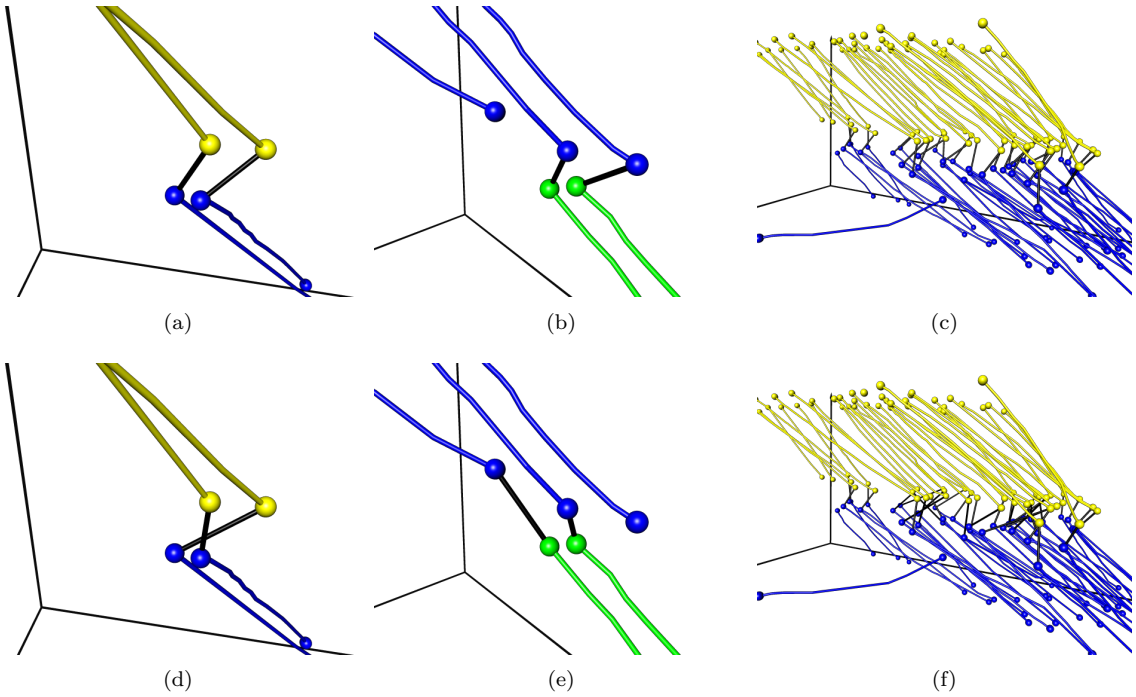


Figure 4.5. Figures (a), (b) and (c) show the connections an expert considered correct. Figures (d), (e) and (f) show the connections computed by a maximum weighted matching with weights Φ_o computed as described in Section 4.4.3.1.

function of $\mathbf{x}$ and abbreviate the probability of $P(\mathbf{x} = x_i)$ by $p(x_i)$. Similarly we write $P(\mathbf{m} = y_m) = p(y_m) = p(m)$ for the discrete random variable. We denote independence of two variables $\mathbf{x}, \mathbf{m}$ by $\mathbf{x} \perp\!\!\!\perp \mathbf{m}$. $\mathbf{P}$ denotes an $M \times N$ matrix that contains the values of a posterior $P(\mathbf{m}|\mathbf{x})$ for all possible pairs x_n, y_m. Furthermore, we write $N_P = \sum_{m,n=1}^{M,N} P(\mathbf{m}|\mathbf{x})$.

The operator $\mathrm{diag}(v)$ creates a diagonal matrix from a vector $v \in R^D$ by $\mathrm{diag}(v) = \begin{pmatrix} v_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & v_D \end{pmatrix}$.

We denote a vector containing only ones as $\mathbf{1}$.

For the point matching, we introduce discrete random variables $\mathbf{x}_i$ each of which corresponds to one $x_i \in X$.

Abbreviations We use the following abbreviations:

ICP	= Iterative closest points
GMM	= Gaussian mixture model
CPD	= Coherent point drift
DCG	= Distance compatibility graph
FN	= False negatives
FP	= False positives
D	= Disagreements between two point matchings
MAP	= Maximum a posteriori
MWM	= Maximum weighted matching for point matching
PGM	= Probabilistic graphical model for point matching

4.3 Previous work on point and line set registration

In this section, we describe previous work on point alignment, registration and matching problem. We give an overview of methods related to ours in Section 4.3.1. We then describe in more detail the

methods we build our work on (Figure 4.4). Section 4.3.2 briefly describes the basics of a distance compatibility graph that can be used for finding an initial coarse alignment. Section 4.3.4 explains the statistical framework used for registration of points sets which we later modify to handle orientations in addition to points. Finally, we introduce the basics of probabilistic graphical model for point matching in Section 4.3.5.

4.3.1 Overview

Computing a transformation given matching point pairs Given point pairs, computing a proper transformation is typically done by first defining a distance measure for corresponding points, usually Euclidean distance, and then solving for the transformation parameters that minimize this distance. Depending on the chosen transformation model, the objective function for minimization must be regularized.

The most simple transformation is the linear transformation $\mathcal{T}(x) = Bx + t$, where B is a matrix, x the coordinate of the transformed point and t a translation vector. If B is a rotation matrix the transformation is called *rigid body transformation*. If $B = sR$ with an isotropic scaling factor s and a rotation matrix R then the transformation is called a *similarity transform*. Eggert et al. [136] summarize and explain the four most common methods to compute similarity transforms. We make use of the results given by Umeyama [137] for computing a similarity transformation given point pairs and describe the idea in more detail in Section 4.3.3. Closely related to this problem is the Procrustes analysis where the residual after alignment is used to analyze variation of shapes (see Gower and Dijksterhuis [138] for an introduction). Walker et al. [139] propose a closed form solution to compute a rigid body transform using dual quaternions. Here, not only coordinates of points but also a unit vector measurements describing an orientation property are taken into account. Linear transformations usually require no regularization, since the number of transformation parameters in low dimensions is small compared to the number of points that are to be transformed.

Often, instead of just transforming the point pairs, one might want to interpolate the domain the points are embedded in by using the displacement vectors induced by the landmarks as reference translations. As Walberg points out [140], this is essentially an interpolation problem of the displacements vectors. In the context of image processing this is called image *warping*.

The most prominent warping technique is the thin plate spline algorithm (TPS) that was introduced by Duchon [141] and investigated further by Wahba [142] and Bookstein [143]. In brief, the interpolation is described with a basis function model where parameters are estimated by minimizing second order derivatives of the function. TPS has become very popular due to smoothness guarantees of the function and the ease of implementation (closed form solution).

Another more recent method for interpolation is the moving least squares algorithm (MLS) which goes back to Backus and Gilbert [144]. MLS computes a deformation that is as close as possible to a rigid transformation locally, meaning that local shearing and scaling are minimized. A nice description of MLS for point and line sets is given by Schaefer et al. [145].

Establishing a coarse alignment Finding a reasonable transformation if the corresponding pairs of the points are yet unknown is a challenging task. For finding a reasonable transformation, typically an error measure is computed and the transformation that yields the lowest error is chosen. However, a measure typically can have local minima in the parameter space that is induced by the transformation parameters. Because the parameter space is often high dimensional (already four-dimensional for a simple linear transformation of two-dimensional points, as described above), it cannot be sampled to find the absolute minimum. To solve this problem, often a subset of points that match is identified instead and the proper transformation is computed from this subset using, for example, one of the methods described in the previous paragraph. Here, Euclidean distance cannot serve for defining an energy functional for optimization of transformation parameters since the points can be arbitrarily shifted. Instead, one must find translation invariant features of the points that discriminate matching pairs from those that do not.

One approach is to use a local feature descriptor, such as the sorted distance to neighbors to measure similarities of local patterns. Li et al. [146] give an overview of and evaluate different techniques for creating a feature descriptor. Matching subsets of points can then be found with

sampling techniques using the feature strength for computing the probability of a match. A popular sampling technique is the Ransac algorithm introduced by Fischler and Bolles [147]. Sampling has been successfully applied in point set registration in biology (see, for example, Preibisch et al. [148]), image registration (Saalfeld et al. [74]) and has also been used to register lines (Yao et al. [149]) and has been shown to be astonishingly stable even in the presence of noise, outliers and strong deformation.

If the point sets are small, the problem can also be converted into finding cliques in a graph representation of the points, called the distance compatibility graph (DCG). This method is often used for molecular shape analysis (for example, Baum et al. [150]) and has also been successfully applied for matching of neuron ends (for example, Dercksen et al. [132]). We use this algorithm to establish a coarse alignment and describe this approach in more detail in Section 4.3.2.

Finding a proper registration after pre-alignment Because the above described methods only make use of a subset of points, the computed transformation can only be an approximation of the best transformation. A common technique for improving a coarse alignment is to make assumption on point pairs and iteratively refining the transformation using these potential pairs.

The most popular algorithm for registration of point sets that are coarsely aligned is the iterative closest points algorithm (ICP). The method in its simplest variant iterates over two steps:

1. For each $x \in X$ find the closest point $y \in Y$.

2. Compute a transformation based on the assigned point pairs.

This method is easy to implement, and convergence to a local minimum is guaranteed (see Besel and McKay [151]). Many variants of ICP have been proposed that speed up convergence and handle outliers. Rusinkiewicz and Levoy [152] give an overview about the most common ICP modifications.

A probabilistic variant of iterative closest points was proposed by several authors (Rangarajan et al. [153], Wells [154]). Here, weights for potential points pairs are computed by assuming that points $x \in X$ are distributed as a Gaussian mixture model (GMM)

$$P(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathcal{T}(y_1, \Theta), \sigma^2) + ... + \mathcal{N}(\mathbf{x}; T(y_m, \Theta), \sigma^2),$$

where $\mathcal{T}$ is a transformation with parameters Θ and the $y \in Y$ are the centroids of the Gaussians. To find good parameters, the maximum-likelihood function $E = \Pi_{x_i \in X} P(x_i)$ is minimized with respect to the transformation parameters using the expectation-maximization algorithm (EM).

This method has many advantages including that the deviation of the assumed GMM can serve as a measure of uncertainty of the result after the algorithm terminates; that the method naturally deals with outliers; and that this method is easy to implement in a few lines of code. This approach has also been adapted for solving elastic deformation models (Jian and Vermuri [155], Myronenko and Song [17]). We base our registration in Section 4.4.2 on the work of Myronenko and Song [17]. We will describe their approach in more detail in Section 4.3.4.

A different but closely related problem to ours is the registration of lines as opposed to just matching line ends. This problem arises, for example, if the most prominent features of images that need to be registered are straight edges (see, for example, Kamgar-Parsi and Kamgar-Parsi [156]). We do not employ this approach since lines in our application are better presented by end coordinate and orientation rather than by a line having a particular length.

Point matching After computing a proper registration, the last task to perform is to find the correct connections of line ends. Here, the problem is to find a *matching* of ends. This problem can be formulated as a graph theoretic problem, where a *matching* in a graph $G = (V, E)$ is defined as the subset of edges $E' \subset E$ so that no two edges are adjacent to the same vertex. If a preference of pairs can be computed (for example, by assuming that Euclidean distance is to be minimized), the edges can be assigned a weight that reflects the belief in the actual pairs. In this case, the task is to compute a matching that maximizes the sum of edge weights. This is known as a *maximum weighted matching* (MWM).

For our application, however, static edge weights might not capture enough prior information on the data and the MWM is not necessarily the correct matching. To allow more flexibility to incorporate knowledge on joint assignments, the problem can be formulated in terms of a probabilistic graphical model (PGM). In brief, the matching is described by a probability distribution

$$P(\mathbf{x}_1, ..., \mathbf{x}_n)$$

where each $\mathbf{x}_i$ is a discrete random variable corresponding to a point in X and having M states where each state corresponds to a point in Y. The sought matching here is the maximum a posteriori (MAP) assignment of the distribution, that is, the joint assignment of $\mathbf{x}_1, ..., \mathbf{x}_n$ having the highest probability. This approach has received a lot of attention (Caetano at al. [157], Sanghavi et al. [158]) and has been successfully applied in many application. Amat et al. [86] applied this algorithm in the context of electron tomography. Here, point matching computation is used to find a proper alignment of series of transmission electron microscopy images using the gold particles on top and bottom of the sample for alignment. We explain this approach in more detail in Section 4.3.5.1. We use a probabilistic graphical model for line end matching (Section 4.4.3), where we primarily build upon the work of Amat et al. [86].

Line ends matching in biology Most closely related to our problem is the task of aligning serial sections of confocal microscopy datasets of neurons. Hogrebe et al. [159] and Dercksen et al. [132] use segmented neuron centerlines to compute the registration of the image data. Our pipeline is similar to both (initial transform, refinement and final end point matching). The major difference to both approaches is that we a) include information on line orientation in all three steps and b) use probabilistic methods to compute results. The latter is needed since electron tomograms have a much lower quality than confocal microscopy data and the uncertainty of results need to be modeled to achieve reliable quality.

4.3.2 Distance compatibility graph

We summarize briefly the method of finding an initial matching for point sets using a distance compatibility graph (DCG) (see, for example, Dercksen et al. [132]). The method identifies similar spatial patterns in X and Y. We define a node in the DCG as all pairs (x, y). Two nodes $((x_i, y_k), (x_j, y_l))$ are connected, if $||x_i - x_j|| - ||y_k - y_l|| < d$, where d is a threshold $\in R^+$. This is called a d-bounded DCG. The memory requirement for building the graph is $(NM)^2$ in the worst case, since there are NM nodes and each of these nodes can be connected to each.

Now consider a maximal connected subgraph (clique) in the DCG. Because all nodes in this subgraph are mutually connected, the distances in the subset of X defined by the clique match the distances in the respective subset of Y. Therefore, the subsets of X and Y have a similar pattern. To find matching points, we must find large cliques in the distance compatibility graph. Note that this method requires that the sought transformation is close to rigid. Significant changes of point distances will cause missing edges in the DCG and clique finding will fail. We describe how we implement this method for line ends in Section 4.4.1.

4.3.3 Computing a similarity transformation given landmarks

Finding a similarity transformation for a dataset requires finding a matrix R, a scaling parameter s and a translation t such that $R^\top R = \mathbf{I}$ and $\det(R) = 1$. The transformation of a point y is given by $\mathcal{T}(y, s, R, t) = sRy + t$. We recapitulate the method described by Umeyama [137] in detail here, because we will make use of the same method in Section 4.4.2.

We can find R, s and t by formulating the problem as a least squares minimization problem. Because we are given fixed pairs, here $M = N$. First of all, the overall error for a rigid transformation for the pairs can be written as

$$E = \sum_{i=1}^{N} ||x_i - sRy_i - t||^2 = \sum_{i=1}^{N} (x_i - sRy_i - t)^\top (x_i - sRy_i - t) \tag{4.1}$$

To find parameters s, R and t that minimize E, we can first eliminate the translation t by multiplying the brackets and computing $\partial E/\partial t$ to obtain

$$t = \frac{1}{N} sR \sum_{i=1}^{N} y_i - \frac{1}{N} \sum_{i=1}^{N} x_i = sR\hat{y} - \hat{x} \tag{4.2}$$

$\hat{y}$ and $\hat{x}$ are the means of the two point sets. Plugging t back into the error function E and denoting $\hat{x}_i = x_i - \hat{x}$ and $\hat{y}_i = y_i - \hat{y}$, we obtain:

$$E = \sum_{i=1}^{N} ||\hat{x}_i - sR\hat{y}_i||^2 \tag{4.3}$$

To simplify this expression, we can again eliminate brackets and write the individual terms in matrix form. The relevant rules for converting the sums to matrices are given in the Appendix A.1.

$$E = \operatorname{tr}(\hat{\mathbf{X}}^\top \hat{\mathbf{X}}) - 2s \operatorname{tr}(\hat{\mathbf{X}}^\top \hat{\mathbf{Y}} R^\top) + s^2 \operatorname{tr}(\hat{\mathbf{Y}}^\top \hat{\mathbf{Y}}) \tag{4.4}$$

Note that R vanishes in the last term due to the constraint $R^\top R = I$. A rotation matrix R minimizing the error function must therefore maximize $\operatorname{tr}(\hat{\mathbf{X}}^\top \hat{\mathbf{Y}} R^\top)$. To find it, we can make use of the following Lemma (see Umeyama [137], Myronenko and Song [17]):

Lemma 4.3.1. *Let R_{DxD} be an unknown rotation matrix and A_{DxD} a known real square matrix. Let $USV^\top$ be a singular value decomposition of A where $UU^\top = VV^\top = I$ and $S = \operatorname{diag}(s_i)$ with $s_1 \geq s_2 \geq ... \geq s_D \geq 0$. Then the optimal rotation matrix R that maximizes $\operatorname{tr}(A^\top R)$ is $R = UCV^\top$ where $C = \operatorname{diag}(1, 1, ..., 1, \det(UV^\top))$.*

This Lemma has been independently developed by different authors. A thorough discussion on its history and a proof for this Lemma to hold in general is given by Myronenko and Song [160]. Using this Lemma, we can compute R first. In a second step, we can compute s by taking the derivative of E (Equation 4.4) with respect to s and solve for s:

$$s = \frac{\operatorname{tr}(\hat{\mathbf{X}}^\top \hat{\mathbf{Y}} R^\top)}{\operatorname{tr}(\hat{\mathbf{Y}}^\top \hat{\mathbf{Y}})} \tag{4.5}$$

t can then be computed by plugging s and R back into to Equation 4.2.

4.3.4 Point set registration using Gaussian mixture models

After having computed a coarse alignment of points, this alignment must be refined further since the method described in Section 4.3.2 only takes into account subsets of the points in each set. We now describe a way to formulate the transformation in terms of a probability distribution. The transformation parameters and distribution parameters can be computed from the distribution by maximizing the likelihood of the data. We describe the approach by Myronenko and Song [17] who use this method to align point sets of arbitrary dimension.

The authors describe a probabilistic variant of ICP. In ICP, point pairs are assigned deterministicly in each iteration. If a point x has many neighbors in Y all of which are close, we might want to compute the transformation based on a weighted average of possible neighbors instead of making a hard assignment to one neighbor and neglect the fact that others are good candidates as well. This weighting of neighbors can be seen as defining a probability distribution for each neighbor y that describes, how likely y is the corresponding point to x. A good transformation $\mathcal{T}$ is then obtained by finding the transformation parameters Θ which maximizes the likelihood of the data.

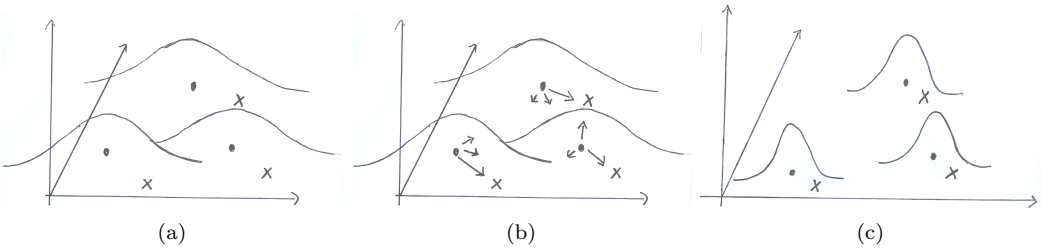
The probability of a point $x \in X$ being a match for $y \in Y$ can be modeled as a joint distribution of a D-dimensional continuous random variable $\mathbf{x}$ and a discrete random variable $\mathbf{m}$ with states $\{y_1, y_2, ..., y_M\}$

$$P(\mathbf{x}, \mathbf{m}) = P(\mathbf{m})P(\mathbf{x}|\mathbf{m}). \tag{4.6}$$

For two sets of two-dimensional points, $P(\mathbf{x}|\mathbf{m})$ could, for example, be an isotropic Gaussian distribution $\mathcal{N}(\mathbf{x}; \mu, \sigma^2)$ with y_m being the centroid μ

$$P(\mathbf{x}|\mathbf{m}) = \mathcal{N}(\mathbf{x}; \mathcal{T}(y_m, \Theta), \sigma^2) \frac{1}{2\pi\sigma^2} \exp(\frac{||\mathbf{x} - T(\Theta, y_m)||^2}{2\sigma^2}) \tag{4.7}$$

where $T(\Theta, y_m)$ denotes a transformation of y_m depending on parameters Θ. Figure 4.6(a) illustrates this distribution. The prior $P(\mathbf{m})$ is uniformly distributed as $1/M$ since we have no measure of certainty of the y_m and must therefore assume each is equally valid.



(a) (b) (c)

Figure 4.6. (a): Illustration of the Gaussian mixture model used for point matching. Dots represent the centroids of the Gaussians which are located at $\mathcal{T}(y_1, \Theta), ..., \mathcal{T}(y_M, \Theta)$. Crosses represent the points in X. (b): Illustration of the posterior computation (E-Step). Arrows indicate values of the posterior for each point in Y. (c): Illustration of the M-step. The computed transformation moves the points a little closer and the deviation of the Gaussians is updated as well.

Optimization $T(\Theta, y_m)$ denotes a transformation of y_m depending on parameters Θ. We would assume that good transformation parameters Θ move y_m closer to an x_n which would increase the probability of the match and also decrease the negative log-likelihood of $\mathbf{x}$ which is given by

$$L(\Theta, \sigma^2) = -\sum_{n=1}^{N} \log \sum_{m=1}^{M} p(m) p(x_n|m) \tag{4.8}$$

A good parameter set Θ can be obtained by minimizing L.

Unfortunately, Equation 4.8 is hard to minimize due to the sum in the log. It is much easier to minimize the log-likelihood of the joint distribution as opposed to the marginal:

$$LL(\Theta, \sigma^2) = -\sum_{m=1}^{M} \sum_{n=1}^{N} p(m) p(x_i|m) \tag{4.9}$$

but we cannot do that either, since we are not given the explicit pairs (x, y). This is a well known problem that can be solved using the expectation maximization algorithm. The key idea of expectation maximization is to minimize the *expectation* of the complete negative log-likelihood function:

$$Q = -\sum_{n=1}^{N} \sum_{m=1}^{M} p^{old}(m|x_n, \Theta^{old}) \log(p(m) p(x_n|m, \Theta)) \tag{4.10}$$

The algorithm reduces Q by iterating two steps:

Expectation First, the posterior $P(\mathbf{m}|\mathbf{x}) = \frac{P(\mathbf{m},\mathbf{x})}{P(\mathbf{x})}$ is computed in the expectation step (E-step). Here, $P(\mathbf{m}, \mathbf{x})$ is given by Equation 4.6 and $P(\mathbf{x})$ can be obtained from $P(\mathbf{m}, \mathbf{x})$ by computing the marginal probability $P(\mathbf{x}) = \sum_{m=1}^{M} P(\mathbf{x}, \mathbf{m})$, which is called a Gaussian mixture model (GMM).

To account for outliers, that is: points that have no match, Myronenko and Song furthermore introduce a term for non matching points $P(\mathbf{x}|M+1) = 1/N$. To assure that the posterior is

normalized, the prior $P(\mathbf{m})$ is weighted with a constant $0 \leq w \leq 1$, so that $P(M+1) = w$ and $P(\mathbf{m}) = (1-w)/M$ for $m \leq M$. The marginal probability of $\mathbf{x}$ then becomes:

$$P(\mathbf{x}) = \sum_{m=1}^{M+1} p(m)P(\mathbf{x}|m) = \frac{1-w}{M} \sum_{m=1}^{M} P(\mathbf{x}|m) + \frac{w}{N} \tag{4.11}$$

Note that we write $P(\mathbf{x})$ although $\int_{-\infty}^{\infty} P(\mathbf{x}) \neq 1$. With this outlier term, the posterior for $P(\mathbf{x}, \mathbf{m})$ in Equation 4.7 takes the form

$$p^{old}(m|x_n) = \frac{\exp^{-\frac{1}{2}} ||\frac{x_n - \mathcal{T}(y_m, \Theta^{old})}{\sigma^{old}}||^2}{\sum_{k=1}^{M} \exp^{-\frac{1}{2}} ||\frac{x_n - T(y_k, \Theta^{old})}{\sigma^{old}}||^2 + \frac{w}{1-w} \frac{(2\pi\sigma^2)M}{N}} \tag{4.12}$$

The posterior for each pair x_i, y_j can be seen as a weight that indicates how much x_i pulls y_j closer. This is illustrated in Figure 4.6(b).

Maximization In the second step, the transformation and distribution parameters are updated by minimizing the expectation of the complete negative log-likelihood Q with respect to Θ and the parameters of the distribution. This step is called the maximization step (M-step). Q is given by Equation 4.10. The first term, p^{old} is the posterior $P(\mathbf{m}|\mathbf{x})$ which was computed in the E-step and stays fixed during optimization. Q is minimized with respect to the new transformation parameters Θ and the parameters of the distribution. Figure 4.6(c) illustrates the update of parameters.

The expectation maximization algorithm always converges to a local minimum (see Bishop et al. [161]). To adapt this algorithm for lines we must 1) define a distribution for the lines, 2) define a transformation model and 3) find a way to compute $\partial Q/\partial\theta = 0$ for each parameter θ in the M-step.

Transformation Myronenko and Song [17] distinguish two cases: All points undergo the same transformation (a) and the transformation differs for each y_m (b).

As for (a), Myronenko and Song describe a solution for both affine transform and similarity transform (Section 4.3.3). For the similarity transform of a two-dimensional point set, $P(\mathbf{x}|\mathbf{m})$ is:

$$P(\mathbf{x}|\mathbf{m}) = \mathcal{N}(\mathbf{x}; \mathcal{T}(y_m, \Theta), \sigma^2) = \frac{1}{2\pi\sigma^2} \exp(\frac{||\mathbf{x} - (sRy_m + t)||^2}{2\sigma^2}) \tag{4.13}$$

The task here is to find R, s, t that minimize Equation 4.10 in the M-step. This can be done in a way very similar to computing a transformation given point pairs as described in Section 4.3.3, because Q becomes:

$$Q(s, R, t, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{m,n=1}^{M,N} p^{old}(y_m|x_n)||x_n - sRy_m - t||^2 + \frac{N_P D}{2} \log\sigma \tag{4.14}$$

where $N_P = \sum_{m,n=1}^{M,N} P(\mathbf{m}|\mathbf{x})$. To obtain an optimal σ^2 and $\{R, s, t\}$, we can first eliminate t like in Section 4.3.3, write Q in matrix form (Appendix A.1) and find the rotation matrix using Lemma 4.3.1. s and σ^2 can again be obtained by taking partial derivatives of Q. The affine transformation can be derived by taking simple derivatives since the transformation matrix for an affine transform is not constraint.

In the second case (b) Y is transformed via a displacement vector $\nu(y)$

$$\mathcal{T}(Y, \nu) = Y + \nu(Y). \tag{4.15}$$

Here, the challenge is to find a function ν that allows enough flexibility to capture the deformation of the data but still maintains the overall neighborhood structure of the points. A regularization for computing the displacement function ν is essential.

Myronenko and Song [17] obtain the vectors by minimizing the regularized Q

$$f(\nu, \sigma^2) = Q(\nu, \sigma^2) + \frac{\lambda}{2}\Phi(\nu), \tag{4.16}$$

where Φ penalizes high frequencies of ν. The calculations for deriving the update equations for parameters in the M-step are rather involved and out of the scope of this dissertation. The resulting vectors $\nu(y)$ are the product of an $M \times M$ smoothing matrix $\mathbf{G}$ and, for 2D points, an $M \times 2$ matrix $\mathbf{W}$ by $\mathcal{T}(y_m, \Theta) = y_m + (\mathbf{G}(m, \cdot)\mathbf{W})^\top$, where $\mathbf{G}(m, \cdot)$ denotes the mth row of $\mathbf{G}$. The transformation parameters Θ here are the entries in the matrix $\mathbf{W}$, whereas $\mathbf{G}$ is initialized once as $g_{ij} = \exp(-\frac{1}{2\beta^2}||y_i - y_j||^2)$ and stays fixed. They call the algorithm for computing the optimal $\mathbf{W}$ the Coherent Point Drift (CPD) algorithm because the results are in accordance to motion coherence theory.

4.3.5 Probabilistic graphical model for point matching

The previously defined distribution gives a likelihood for each point pair to occur. However, we cannot simply assign each point in X to its most likely counterpart in Y because assignments might conflict and the final pairs have to be unique. To handle these cases, a model must include the constraint that the assignment is a *matching* (Section 4.3.1). Furthermore, the assignments of neighboring points might influence our decision on an assignment. For example, we would expect that the assignment of line ends in Figure 4.7(a) (left side) is preferable to the assignment in Figure 4.7(b) (left side) because we expect remaining deformations of the data to be coherent in a small neighborhood of a point.

This problem can again be modeled as a probability distribution describing the point correspondences, though this time the distribution is a joint distribution of all nodes in one set. We explain the construction of this distribution in Section 4.3.5.1. We will see that the best possible matching can be found by computing the maximum a posteriori (MAP) assignment of the distribution. Because finding the MAP assignment is an NP-hard problem in general, an approximation algorithm must be implemented to tackle this problem. One such algorithm is belief propagation which we describe in Section 4.3.5.2.

Details on computing a point matching with a probabilistic graphical model are given by Koller and Friedman [162] and Amat et al. [86].

4.3.5.1 Probabilistic formulation of the point matching problem

Distribution Nodes in X are represented by discrete random variables $\mathbf{x}_1, ..., \mathbf{x}_N$, each of which can have $M + 1$ states. Each $\mathbf{x}_i$ must be assigned to either one node in Y or to the dummy node y_0 with the constraint that no two $\mathbf{x}_i, \mathbf{x}_j$ can be assigned to the same node in Y. Figure 4.7 shows some possible states for a joint distribution of three variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, corresponding to three endpoints of lines (green). The possible states Y_1, Y_2 of the variables correspond to two lines in the next section (blue). Y_0 is the dummy node. For example, in Figure 4.7(a), a connection of x_2, y_1 and x_3, y_2 induces the assignment $\mathbf{x}_2 = y_1, \mathbf{x}_3 = y_2, \mathbf{x}_1 = y_0$. Note that the dummy node is the only node that can be assigned to more than one partner, as illustrated in Figure 4.7(d).

To find a good assignment, we must first define the joint distribution $P(\mathbf{x}_1, ...\mathbf{x}_n)$ and then find the joint assignment that yields the largest probability.

$P(\mathbf{x}_1, ..., \mathbf{x}_n)$ is defined by a Gibbs distribution (Koller and Friedman [162])

$$P(\mathbf{x}_1, ..., \mathbf{x}_N) = \frac{1}{Z} \prod_{i=1}^{N} \Phi_o(\mathbf{x}_i) \prod_{i,j=1}^{N,N} \Phi_{oo}(\mathbf{x}_i, \mathbf{x}_j), \tag{4.17}$$

each state of which is a valid matching. Z is a normalization term that assures that $P(\mathbf{x}_1, ..., \mathbf{x}_N)$ is a valid distribution. The Φs are called *factors* and can be thought of as a matrix containing weights corresponding to the beliefs we have about a particular assignment. $\Phi_o(\mathbf{x}_i)$ is a *singleton* factor containing information on how likely $\mathbf{x}_i$ and each $y_1, .., y_m, y_0$ match if no further information is available. The *pair* factors $\Phi_{oo}(\mathbf{x}_i, \mathbf{x}_j)$ represent beliefs about the joint assignment $\mathbf{x}_i = y_k$ and $\mathbf{x}_j = y_l$. We can use the pair factors to model, for example, the mutual exclusiveness constraint of the assignments.

The product operation for factors is depicted in Figure 4.8(a). The result of a factor product is a new factor describing a belief in the joint assignments of the variables in the two factors that are multiplied. Each entry is computed by multiplying the entries consistent with the assignment

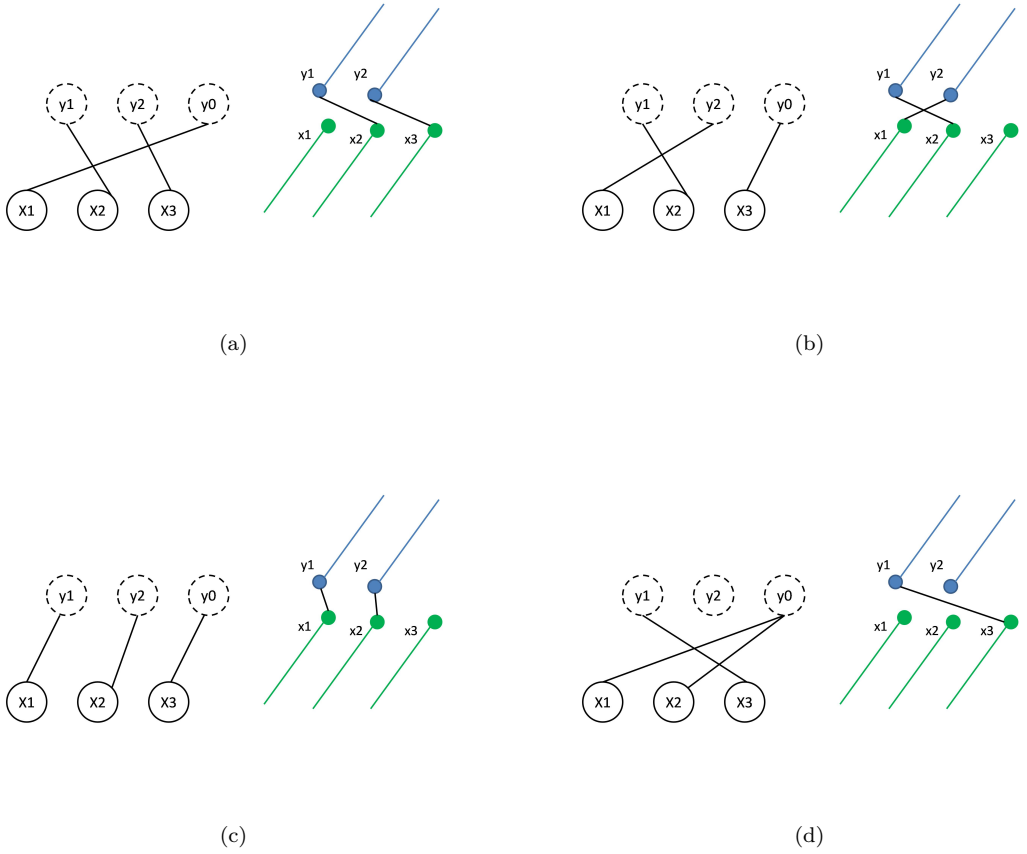


Figure 4.7. The green and blue lines (right illustration) illustrate lines that are to be matched. The random variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and their states corresponding to the line connections are sketched on the left side. Here, an assignment for a variable $\mathbf{x}$ is indicated by a dashed line.

of this entry. For example, in Figure 4.8(a), the entry $A = 1, B = 1$ of the resulting factor is computed by multiplying $A = 1, B = 1$ of the pair factor and $A = 1$ of the singleton factor.

Factor values In Amat et al. [86] the values of the singleton factors are computed by

$$\Phi_o(\mathbf{x}_i) = NCC(x_i, y_j)e^{-\lambda_c||x_i - y_j||},$$

where NCC denotes the normalized cross-correlation of an image patch around both x_i and y_j and λ_c is a parameter weighting the influence of the Euclidean distance of x_i and y_j.

Amat et al. [86] fill the pair factors by considering the displacement vectors $y_k - x_i$, $y_l - x_j$ induced by a joint assignment $\mathbf{x}_i = y_k, \mathbf{x}_j = y_l$. The distance $\mathrm{d_s}(x_i, y_k, x_j, y_l) = ||(y_k - x_i) - (y_l - x_j)||$ is used as a means to infer how coherent two assignments are. If $\mathrm{d_s}$ is large, then the displacement vectors disagree. For example, in Figure 4.7(b), $\mathrm{d_s}(x_1, y_2, x_2, y_1)$ would be larger than $\mathrm{d_s}(x_1, y_1, x_2, y_2)$ in Figure 4.7(c), because the connections cross and the displacement vectors are far apart. To prevent such joint assignments, large values of $\mathrm{d_s}$ are penalized by filling Φ_{oo} with

$$\Phi_{oo}(\mathbf{x}_i = y_k, \mathbf{x}_j = y_l) = \lambda_s \exp(-\lambda_s \, \mathrm{d_s}(x_i, y_k, x_j, y_l)), \tag{4.18}$$

To account for the mutual exclusiveness constraint, entries in Φ_{oo} can be set to 0 if $k = l$ (see Koller and Friedman [162]). Amat et al. [86] furthermore restrict potential assignment to a variable. They

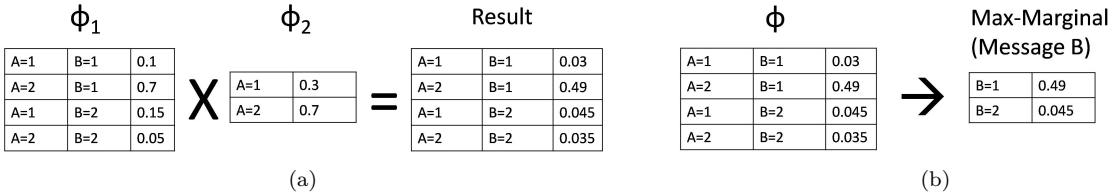


Figure 4.8. (a) illustrates the product operation for factors. Each entry in the resulting factor represents a joint assignment $A = i, B = j$. To compute an entry, entries in Φ_1 and Φ_2 that are coherent with the corresponding assignment of the result entry must be multiplied. (b) shows an example of a max marginal for the variable B. The resulting factor contains the maximal value for each assignment of B in factor Φ. The result can be used as a message in belief propagation.

define a threshold on the distance $\mathrm{d}(x_i, y_j) = ||x_i - y_j||$ of two points and y_j can only be assigned to $\mathbf{x}_i$ if $t_d \geq \mathrm{d}(x_i, y_j)$.

4.3.5.2 Belief propagation for finding the maximum a posteriori assignment

Each valid matching is a single joint assignment in the distribution $P(\mathbf{x}_1, ...\mathbf{x}_N)$. We now want to compute $\mathrm{argmax}_{\mathbf{x}_1,...\mathbf{x}_N} P(\mathbf{x}_1, ...\mathbf{x}_N)$, the joint assignment with the largest probability. This is called the maximum a posteriori (MAP) assignment. Computing a MAP assignment is not generally feasible due to the NP-hardness of the problem. To see this, consider that the number of possible joint assignments in the network is $v(\mathbf{x}_1) \cdot v(\mathbf{x}_2) \cdot ... \cdot v(\mathbf{x}_n)$, where $v(\mathbf{x})$ denotes the number of possible assignments for variable $\mathbf{x}$.

Inference in a graphical model, that is, finding a MAP assignment or marginals of variables for a probabilistic graphical model is an important research topic. A huge variety of algorithms has been proposed for approximate inference on graphical models. Most of these algorithms exploit particular properties of the underlying distribution or are designed to guarantee a specific error bound (see, for example, Komodakis and Tziritas [163]). We use an approximation technique called belief propagation on a graph representation called a factor graph (see Koller and Friedman [162], Amat et al. [86] and Kschischang et al. [164]). An exhaustive reference for details on this algorithm is given by Koller and Friedman [162].

Belief propagation is an iterative algorithm that updates a current belief about the best assignment in each step, until the belief converges to a local maximum. In brief, each node in a factor graph represents one of the factors as depicted in Figure 4.9. Nodes are connected if their factors share a variable. Messages expressing the belief about a MAP assignment of a variable are passed by the nodes in a particular order (red arrows and numbers in Figure 4.9). Each node uses incoming messages and its own factor value to compute its own belief. This is in turn passed on to the next nodes as an incoming message. For example, in Figure 4.9, the node representing the pair factor for $\mathbf{x}_1, \mathbf{x}_2$ uses message one to compute a belief on the assignment on $\mathbf{x}_2$ and passes this belief on to the node representing the singleton factor for $\mathbf{x}_2$ in message 5. The message passing finishes if all messages agree. The MAP assignment can be read from the messages in the graph after convergence. See Koller and Friedmann [162] for details on this algorithm. Belief propagation on a factor graph can be used to determine the marginal of variables, in which case it is called the *sum-product algorithm*, or to infer a MAP assignment, then it is called the *max-sum algorithm*.

To compute the messages for the latter case, we first note that the MAP assignment is defined as

$$\mathrm{argmax}_{x_1,x_2,...,x_n} \prod_{i=1}^{N} \Phi_o(x_i) \prod_{i,j=1}^{N,N} \Phi_{oo}(x_i, x_j) \tag{4.19}$$

To compute a message, each factor first computes the factor product of incoming messages. Because

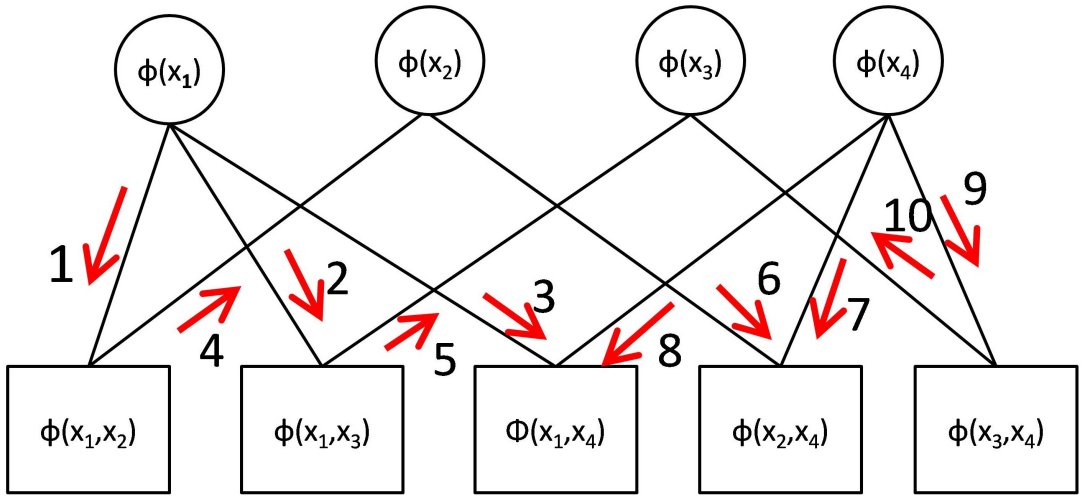


Figure 4.9. Example of a factor graph. Circles depict singleton factors, squares pair factors. Red arrows and numbers indicate an exemplary message passing schedule.

we can rewrite Equation 4.19 in terms of the log of the distribution

$$\text{argmax}_{x_1,x_2,\ldots,x_n} \sum_{i=1}^{N} \log \Phi_o(x_i) \sum_{i,j=1}^{N,N} \log \Phi_{oo}(x_i, x_j), \tag{4.20}$$

in practice, we do not evaluate the product of the factors but the sum of the log, which is why the algorithm for MAP assignment is called the *max-sum-algorithm*.

In a second step, the *max-marginal belief* of this factor product is computed. An example for a max-marginal belief is depicted in Figure 4.8(b). Entries in a max-marginal factor are just the largest assignment values for each state and the factor represents the current belief about a variable. This is the message that is then passed on to the next factors.

Many details of this simple algorithm are subject to intensive research: The factor graph representation (sometimes also called the Bethe graph) is only one possible representation. Factors can also be organized in cluster graphs. In this case there are many valid ways to organize factors in nodes and to draw edges between the clusters. Another issue is the message scheduling. A node in a factor graph can only pass a message on if all incoming messages have arrived. However, the user is free to choose the order in which messages are passed. Graph layout and message scheduling can have a significant impact on the performance of belief propagation. A thorough discussion on this topic and a discussion on the most common algorithm used is in Koller and Friedman [162].

If after a pass on the network all messages agree, belief propagation is said to have converged. The graph is then called *maximum calibrated*. After belief propagation has terminated, the MAP assignment can be read from the messages in the graph.

Unfortunately, convergence to a maximum is not guaranteed. Success depends on the underlying model. If two assignments conflict, messages with different beliefs about MAP assignments will be passed back and forth between the nodes and the algorithm will reach no conclusion. This oscillatory behavior is often a local problem of a network (see Koller and Friedman [162], 401 ff, 570 ff). In Section 4.4.3.2, we will address how we deal with non-converging networks when we match line ends.

4.4 Microtubule end stitching

In this section we present our contributions. The previously described algorithms were used to align, register and match point sets. We adapt these algorithms to handle both points and orientations.

4.4.1 Distance compatibility graph for matching microtubule ends

We compute an initial alignment by finding cliques in a distance compatibility graph as described in Section 4.3.2. using the line ends as point sets. In contrast to the work of Dercksen et al. [132], we further restrict connections of nodes in the DCG by an edge by taking into account the orientation of the lines as well. We only connect nodes by an edge if $|\arccos(\vec{x}_i \cdot \vec{x}_j) - \arccos(\vec{y}_k \cdot \vec{y}_l)| < \alpha$, where $\cdot$ denotes the inner product of the unit vectors that describe the line orientations. Because in a clique in the DCG the mutual angles agree, the patterns on a unit sphere formed by the orientation vectors must also agree, which is, in addition to mutual distances of end coordinates, another strong indicator that the lines match. With this modification, we find cliques using the Bron-Kerbosch algorithm [165] similar to Dercksen et al. [132]. Due to the high memory requirement of $(NM)^2$ in the worst case and the exponential running time of the Bron-Kerbosch algorithm, the DCG graph should not contain more than approximately $10,000$ nodes. This means that the number of vertices in X and Y should be smaller than 100. This is much fewer than the number of endpoints of microtubule centerlines in subsequent tomograms. We therefore use a simple heuristic to reduce the number of points: First, we sort all lines in descending order of angle of the line with the x,y-plane. Then, we pick the first 50-100 points in each list. Furthermore, we only consider cliques if their size is at least $10 - 30\%$ of the number of nodes in the smaller set. We only consider the first 1000 cliques computed by the Bron-Kerbosch algorithm.

From the resulting point matching, we compute an optimal (in a least square sense) rigid alignment on the form $\mathcal{T}(x, \Theta) = Rx + t$ using a singular value decomposition as described by Umeyama [137] and Myronenko and Song [160]. To compute this alignment, we only take into account the coordinates of the line ends. A method to employ line orientations by formulating the transformations with dual quaternions is described by Walker et al. [139]. However, for our goal, the described alignment is sufficient, because it is only the first step in the alignment pipeline and the resulting transformation will be refined further, as described in the next section.

4.4.2 Registration of line ends with expectation maximization

The previously described DCG alignment takes into account only a small subset of lines to reduce running time. Consequently, depending on the error measure, the resulting transformation $\mathcal{T}(Y, \Theta)$ does not necessarily induce an optimal alignment of the line ends. To refine the transformation, we modify the probabilistic registration algorithms described in Section 4.3.4 to also take the orientation associated to each point into account. We developed algorithms for the computation of three types of transformations:

1) A rotation of the line sets. We show how the orientations of the lines can be registered by computing a rotation matrix $R \in \mathcal{R}^{3\times3}, \mathcal{T}(\vec{y}, \Theta) = R\vec{y}$. Because we assume that all lines end on a plane, the rotation matrix models a rotation around the z-axis, so that

$$\mathcal{T}(\vec{y}, \Theta) = R\vec{y} = \begin{pmatrix} \tilde{R} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix} \vec{y}. \tag{4.21}$$

Using orientations only to compute the best rotation of the line set is useful if the coordinates of the line ends do not provide enough information to decide on reasonable transformation parameters but the orientations do. We will encounter examples in the result section.

2) A similarity transform for the coordinates. The coordinates, in contrast to the orientations, can also be scaled and translated. We compute a similarity transform $\mathcal{T}(y, \Theta) = s\tilde{R}y + t$, where $\tilde{R} \in \mathcal{R}^2$ again is a rotation matrix that rotates the line ends on the plane, $s \in \mathcal{R}$ is a scaling parameter and $t \in \mathcal{R}^2$ is a translation vector. Because the orientations are three-dimensional vectors, we construct the rotation matrix for orientations from $\tilde{R}$ using Equation 4.21. Orientations are rotated again using $\mathcal{T}(\vec{y}, \Theta) = R\vec{y}$.

3) An elastic deformation. The transformation here only applies to the coordinates and is given by individual translation vectors $\nu(y)$ so that $\mathcal{T}(y, \Theta) = y + \nu(y)$.

4.4.2.1 Algorithm 1: Optimal rotation using line orientation only

We first register only the orientations of the lines. We define the transformation $\mathcal{T}$ to be a rotation $\mathcal{T}(\vec{y}, \Theta) = R\vec{y}$ where R induces a rotation around the z-axis (Equation 4.21). To do so, we introduce

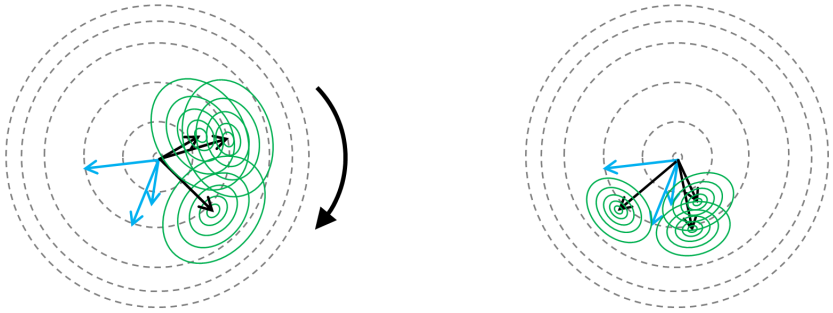
two more random variables $\vec{\mathbf{x}}$ (continuous) and $\vec{\mathbf{m}}$ (discrete with states $\vec{y}_1, ...\vec{y}_m$) for the orientations of lines and describe the joint distribution of these variables by

$$P(\vec{\mathbf{x}}, \vec{\mathbf{m}}) = P(\vec{\mathbf{m}})P(\vec{\mathbf{x}}|\vec{\mathbf{m}}) \tag{4.22}$$

Because $\vec{\mathbf{x}}$ is periodic, we cannot simply assume that $P(\vec{\mathbf{x}}|\vec{\mathbf{m}})$ is distributed as a Gaussian because the unit vectors are periodic. The distribution should measure how close two directions $\vec{x}, \vec{y}$ are. A natural way to do this, is to measure the angle between the two vectors since the directions in our case have no particular length. The Fisher-Mises distribution describes a Gaussian-like unimodal distribution for three-dimensional periodic random variables (see Mardia [166]):

$$P(\vec{\mathbf{x}}|\vec{\mathbf{m}}) = \mathcal{F}(\vec{\mathbf{x}}; \vec{y}_m, \kappa) = \frac{\kappa}{2\pi(e^{\kappa} - e^{-\kappa})} \exp(\kappa \mathcal{T}(\vec{y}_m, \Theta)^{\top} \vec{\mathbf{x}}). \tag{4.23}$$

The Fisher-Mises distribution is equivalent to a Gaussian distribution on a sphere, as illustrated in Figure 4.10(a). κ is the equivalent to σ^2 and is referred to as the concentration parameter. The prior $P(\vec{\mathbf{m}})$ is uniformly distributed as $1/M$ as in the work of Myronenko and Song [17].



(a) (b)

Figure 4.10. (a) Illustration of expectation maximization for periodic variables. Arrows indicate the orientation unit vectors depicted here as a projection to the plane. The green circles depict values of the Fisher-Mises distribution with its center located at the arrow tip. (b) In the M-step a rotation around the center is computed, that moves arrow tips closer. Simultaneously, the concentration parameter κ is updated.

To obtain the optimal rotation using expectation maximization, we must compute the posterior and minimize the negative complete log-likelihood function Q (Equation 4.10) with respect to R and κ.

E-step The posterior $P(\vec{\mathbf{m}}|\vec{\mathbf{x}})$ is computed by

$$P(\vec{\mathbf{m}}|\vec{\mathbf{x}}) = \frac{P(\vec{\mathbf{m}}, \vec{\mathbf{x}})}{P(\vec{\mathbf{x}})}. \tag{4.24}$$

$P(\vec{\mathbf{m}}, \vec{\mathbf{x}})$ can be computed from the prior $P(\vec{\mathbf{m}}) = 1/M$ and $P(\vec{\mathbf{x}}|\vec{\mathbf{m}})$ which is given by Equation 4.23. With the outlier term $P(\vec{\mathbf{x}}|M+1) = 1/N$ and weighting the priors with $0 \leq w \leq 1$ (Section 4.3.4) the joint distribution becomes:

$$P(\vec{\mathbf{m}}, \vec{\mathbf{x}}) = P(\vec{\mathbf{m}})P(\vec{\mathbf{x}}|\vec{\mathbf{m}}) = \frac{1-w}{M} \frac{\kappa}{2\pi(e^{\kappa} - e^{-\kappa})} \exp(\kappa (R\vec{y}_m)^{\top} \vec{\mathbf{x}}), \text{ for all } m \leq M \text{ and} \tag{4.25}$$

$$P(\vec{\mathbf{x}}, M+1) = w/N, \text{ for } m = M+1. \tag{4.26}$$

$P(\vec{\mathbf{x}})$ can be computed from $P(\vec{\mathbf{m}}, \vec{\mathbf{x}})$ by marginalizing out $\vec{\mathbf{m}}$:

$$\begin{aligned} P(\vec{\mathbf{x}}) &= P(\vec{\mathbf{x}}, M+1) + \sum_{m=1}^{M} P(\vec{\mathbf{m}}, \vec{\mathbf{x}}) \\ &= \frac{w}{N} + (1-w) \sum_{k=1}^{M} \frac{\kappa}{2M\pi(e^{\kappa} - e^{-\kappa})} \exp(\kappa (R\vec{y}_k)^{\top} \vec{\mathbf{x}}) \end{aligned} \tag{4.27}$$

The posterior becomes (4.25 and 4.27):

$$P(\vec{\mathbf{m}}|\vec{\mathbf{x}}) = \frac{P(\vec{\mathbf{m}}, \vec{\mathbf{x}})}{P(\vec{\mathbf{x}})} = \frac{\exp(\kappa \vec{x}_n^{\top} R \vec{y}_m)}{\sum_{k=1}^{M} \exp(\kappa \vec{x}_n^{\top} R \vec{y}_k) + \frac{wM}{(1-w)N} \frac{2\pi(\exp(\kappa) - \exp(-\kappa))}{\kappa}} \tag{4.28}$$

M-step In the M-step, we must write out the expectation of the negative log-likelihood function Q (Equation 4.10) and minimize it with respect to the transformation parameters R and the distribution parameter κ. Q becomes:

$$\begin{aligned} Q(R, \kappa) = -\kappa \sum_{m,n=1}^{M,N} p^{old}(m|\vec{x}_n) \vec{x}_n^{\top} R \vec{y}_m - N_P \log(\frac{\kappa}{e^{\kappa} - e^{-\kappa}}) + \text{const.} = \\ -\kappa \operatorname{tr}((\vec{\mathbf{X}}^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}})^{\top} R) - N_P \log(\frac{\kappa}{e^{\kappa} - e^{-\kappa}}) + \text{const.} \end{aligned} \tag{4.29}$$

We split the term $\operatorname{tr}((\vec{\mathbf{X}}^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}})^{\top} R)$ into terms dependent and independent of $\tilde{R}$:

$$\operatorname{tr}((\vec{\mathbf{X}}^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}})^{\top} R) = \operatorname{tr}((\vec{\mathbf{X}}_{x,y}^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}}_{x,y})^{\top} \tilde{R}) + \vec{\mathbf{X}}_z^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}}_z \tag{4.30}$$

where $\vec{\mathbf{X}}_{x,y}$ is a matrix that contains the first two columns of $\vec{\mathbf{X}}$ and $\vec{\mathbf{X}}_z$ contains the last column of $\vec{\mathbf{X}}$.

The optimal rotation can be obtained with a singular value decomposition from the term $\vec{\mathbf{X}}_{x,y}^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}}_{x,y}$ (equivalently to Myronenko and Song [17], see Umeyama [137] and Myronenko and Song [160] for details on the method). κ cannot be obtained in closed form but it can be computed with Newton's Method. We iterate

$$\kappa = \kappa^{old} - \frac{\partial Q}{\partial \kappa} / \frac{\partial^2 Q}{\partial^2 \kappa} \text{ until convergence, where}$$

$$\frac{\partial Q}{\partial \kappa} = -\operatorname{tr}((\vec{\mathbf{X}}_{x,y}^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}}_{x,y})^{\top} \tilde{R}) - \operatorname{tr}(\vec{\mathbf{X}}_z^{\top} \mathbf{P}^{\top} \vec{\mathbf{Y}}_z) - N_P(\frac{1}{\kappa} - \coth \kappa), \text{ and} \tag{4.31}$$

$$\frac{\partial^2 Q}{\partial^2 \kappa} = -N_P(-\frac{1}{\kappa^2} + \frac{1}{\sinh \kappa^2}). \tag{4.32}$$

Because we only compute a rotation, the coordinates are not aligned. To align them, we first note that, even though we did not explicitly take the coordinates into account, the likelihood that two orientations match also applies to the corresponding line end coordinates. To compute an alignment of the coordinates, we compute the squared distances of the points, which we weight by the posterior computed from the obtained optimal parameters Θ^*, $P(\vec{\mathbf{m}}|\vec{\mathbf{x}}, \Theta^*)$:

$$S = \sum_{m,n=1}^{M,N} p(\vec{y}_m|\vec{x}_n, \Theta^*)|x_n - (\tilde{R}y_m + t)|^2.$$

To minimize the distances, we first eliminate the brackets and take the derivative with respect to t:

$$S = \sum_{m,n=1}^{M,N} p(\vec{y}_m|\vec{x}_n, \Theta^*)(-2x_n + 2\tilde{R}y_m + 2t).$$

Solving for t yields:

$$t = \frac{1}{N_P} \sum_{m,n=1}^{M,N} p(\vec{y}_m|\vec{x}_n, \Theta^*) x_n - \frac{1}{N_P} \sum_{m,n=1}^{M,N} p(\vec{y}_m|\vec{x}_n, \Theta^*) \tilde{R} y_m = \mu_x - \tilde{R}\mu_y.$$

We summarize the algorithm in Box 4.1.

Rotation from Orientations:

- Initialization: $R = \mathbf{I}, 0 \leq w \leq 1, 0 < \kappa \ll 1$
- Repeat until convergence:
  - E-step: Compute posterior matrix $\mathbf{P}$. $\mathbf{P}_{nm} = \frac{\exp(\kappa \vec{x}_n^\top R \vec{y}_m)}{\sum_{k=1}^{M} \exp(\kappa \vec{x}_n^\top R \vec{y}_k)) + \frac{wM}{(1-w)N} \frac{2\pi(\exp(\kappa) - \exp(-\kappa))}{\kappa}}$
  - M-step: Solve for $\tilde{R}$ and κ
    - $N_P = \mathbf{1}^\top \mathbf{P} \mathbf{1}$,
    - $A = \vec{\mathbf{X}}_{2d}^\top \mathbf{P} \vec{\mathbf{X}}_{2d}$, compute SVD of $A = USV^\top$,
    - $\tilde{R} = UCV^\top$, where $C = \mathrm{diag}(1, ..., \det(UV^\top))$,
    - iterate $\kappa = \kappa - \frac{-\mathrm{tr}((\vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d})^\top \tilde{R}) - \mathrm{tr}(\vec{\mathbf{X}}_z^\top \mathbf{P}^\top \vec{\mathbf{Y}}_z) - N_P(\frac{1}{\kappa} - \coth \kappa)}{-N_P(-\frac{1}{\kappa^2} + \frac{1}{\sinh \kappa^2})}$, until convergence,
    - $R = \begin{pmatrix} \tilde{R} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix}$.
- Compute orientations: $\vec{\mathbf{Y}}^{final} = \vec{\mathbf{Y}} R^\top$.
- Compute coordinates: $\mu_x = \frac{1}{N_p} \mathbf{X}^\top \mathbf{P}^\top \mathbf{1}, \mu_y = \frac{1}{N_p} \mathbf{Y}^\top \mathbf{P} \mathbf{1}, t = \mu_x - \tilde{R}\mu_y, \mathbf{Y}^{final} = \mathbf{Y}\tilde{R}^\top + \mathbf{1}t^\top$.

Box 4.1. Algorithm for computing an optimal rotation for orientation vectors.

4.4.2.2 Algorithm 2: Optimal rotation, scale and translation using end point coordinates and orientations of lines

For an accurate alignment of the end coordinates, the previous algorithm might not suffice since only orientations are taken into account and the transformation model does not include scaling. In this paragraph, we describe an algorithm that considers both orientations and end coordinates of the lines. We compute a similarity transform for the end point coordinates of lines such that $\mathcal{T}(y, \Theta) = s\tilde{R}y + t$. The orientations of lines are again rotated by $\mathcal{T}(\vec{y}, \Theta) = Ry$. The relation of R and $\tilde{R}$ is given by Equation 4.21. To apply the above described framework, we define a joint distribution $P(\mathbf{x}, \mathbf{m}, \vec{\mathbf{x}}, \vec{\mathbf{m}})$ that factorizes as follows:

$$P(\mathbf{x}, \mathbf{m}, \vec{\mathbf{x}}, \vec{\mathbf{m}}) = P(\mathbf{x}|\mathbf{m}) P(\vec{\mathbf{x}}|\vec{\mathbf{m}}) P(\mathbf{m}, \vec{\mathbf{m}}) \tag{4.33}$$

$P(\mathbf{x}|\mathbf{m})$ and $P(\vec{\mathbf{x}}|\vec{\mathbf{m}})$ are defined by Equation 4.7 and 4.23 respectively. We define the prior $P(\mathbf{m}, \vec{\mathbf{m}})$ by

$$P(\mathbf{m}, \vec{\mathbf{m}}) = \frac{1}{M} \delta_{\mathrm{ind}(\mathbf{m}), \mathrm{ind}(\vec{\mathbf{m}})} \tag{4.34}$$

where ind(.) denotes the index operator, which returns the index of the current assignment and δ is the Kronecker delta which is 1 if $\mathrm{ind}(\mathbf{m}) = \mathrm{ind}(\vec{\mathbf{m}})$ and otherwise 0. We chose this prior because we know the pairs $(y_j, \vec{y}_j)$ and that pairs with mixed indices $(y_i, \vec{y}_j), i \neq j$ do not occur.

We must again write out Q for this distribution and transformation model and then solve for $s, \tilde{R}, t, \sigma^2$ and κ that minimize Q in the M-step.

E-step The joint distribution of $\mathbf{x}, \vec{\mathbf{x}}$ and $\mathbf{m}, \vec{\mathbf{m}}$ is (Equation 4.7 and Equation 4.23):

$$P(\mathbf{x}, \vec{\mathbf{x}}, \mathbf{m}, \vec{\mathbf{m}}) = \mathcal{N}(\mathbf{x}; \mathbf{m}, \sigma^2) \mathcal{F}(\vec{\mathbf{x}}; \vec{\mathbf{m}}, \kappa) \frac{1}{M} \delta_{\mathrm{ind}(\mathbf{m}), \mathrm{ind}(\vec{\mathbf{m}})} \tag{4.35}$$

To compute the posterior $P(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}})$, we must again apply Bayes' rule. We add an outlier term like in the previous algorithm. The posterior then is:

$$\begin{aligned} P(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}}) &= \frac{P(\mathbf{x}, \vec{\mathbf{x}}, \mathbf{m}, \vec{\mathbf{m}})}{P(\mathbf{x}, \vec{\mathbf{x}})} = \\ &\frac{(1-w)\mathcal{N}(\mathbf{x}; \mathbf{m}, \sigma^2)\mathcal{F}(\vec{\mathbf{x}}; \vec{\mathbf{m}}, \kappa)\frac{1}{M}\delta_{\mathrm{ind}(\mathbf{m}),\mathrm{ind}(\vec{\mathbf{m}})}}{\frac{w}{N} + \sum_{k=1}^{M}\mathcal{N}(\mathbf{x}; y_k, \sigma^2)\mathcal{F}(\vec{\mathbf{x}}; \vec{y}_k, \kappa)\frac{1}{M}\delta_{k,k}} = \\ &\frac{-\exp(\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_m - t||^2)\exp(\kappa\vec{x}_n^\top R\vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_k - t||^2)\exp(\kappa\vec{x}_n^\top R\vec{y}_k)) + \frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}} \end{aligned} \tag{4.36}$$

M-step The expectation of the complete negative log-likelihood function is given by

$$\begin{aligned} Q(s, \tilde{R}, t, \sigma^2, \kappa) = &+\frac{1}{2\sigma^2}\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)||x_n - s\tilde{R}y_m - t||^2 + N_P \log\sigma \\ &-\kappa\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)\vec{x}_n^\top R\vec{y}_m - N_P\log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.} \end{aligned} \tag{4.37}$$

To solve for each of the parameters during the M-Step, we first compute t by taking the partial derivative of Q with respect to t, as we did in the last section, to obtain (equivalently to Myronenko and Song [17])

$$t = \frac{1}{N_P}\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)x_n - \frac{1}{N_P}\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)\tilde{R}y_m = \mu_x - s\tilde{R}\mu_y \tag{4.38}$$

We then plug t back into Equation 4.37. The log-likelihood becomes:

$$\begin{aligned} Q(s, \tilde{R}, \sigma^2, \kappa) = &\frac{1}{2\sigma^2}\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)||\hat{x}_n - s\tilde{R}\hat{y}_m||^2 + \frac{N_P}{2}\log\sigma \\ &-\kappa\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)\vec{x}_n^\top R\vec{y}_m - N_P\log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.}, \end{aligned} \tag{4.39}$$

where $\hat{x}_n = x_n - \mu_x$ and $\hat{y}_m = y_m - \mu_y$. We then write out $||\hat{x}_n - s\tilde{R}\hat{y}_m||^2$ as a product $(\hat{x}_n - s\tilde{R}\hat{y}_m)^\top(\hat{x}_n - s\tilde{R}\hat{y}_m)$, eliminate the brackets and write Q in matrix form (Appendix A.1):

$$\begin{aligned} Q(s, \tilde{R}, \sigma^2, \kappa) = &\frac{1}{2\sigma^2}(\mathrm{tr}(\hat{\mathbf{X}}^\top \mathrm{diag}(\mathbf{P}^\top\mathbf{1})\hat{\mathbf{X}}) - 2s\cdot\mathrm{tr}((\hat{\mathbf{X}}^\top\mathbf{P}^\top\hat{\mathbf{Y}})^\top\tilde{R}) + s^2\,\mathrm{tr}(\hat{\mathbf{Y}}^\top\mathrm{diag}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}})) - \\ &-\kappa\,\mathrm{tr}((\vec{\mathbf{X}}^\top\mathbf{P}^\top\vec{\mathbf{Y}})^\top R) + N_P\log\sigma^2 - N_P\log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.} \end{aligned} \tag{4.40}$$

At this point we cannot again make use of Lemma 4.3.1 to find the optimal rotation matrix, because the two rotation matrices are related via Equation 4.21 and therefore, the traces would have to be summed, which is not possible because of σ^2, s and κ. Thus, we cannot find a closed form solution for any of the parameters. Instead we must find the optimum for the parameters numerically, except for t, which we have eliminated already. In general, using numerical optimization in this case is problematic, because of the constraint $\tilde{R}^\top\tilde{R} = \mathbf{I}$ and $\det(\tilde{R}) = 1$. In our case, however, the rotation matrix is only two dimensional and can therefore be parameterized by a single angle ρ that can be treated as one additional parameter. To find a numerical solution for this problem, usually first and second derivatives of the objective function are required. These are not hard to find from Equation 4.40, but it is tedious work. For the convenience of the reader, we give details on the derivative with respect to ρ in Section A.2 and list all first and second order derivatives in Table A.1. Box 4.2 summarizes the steps.

Similarity transform:

- Initialization: $R = \mathbf{I}, s = 1, t = \mathbf{0}, 0 \leq w \leq 1, \sigma^2 = \frac{1}{NM}\sum_{m,n=1}^{M,N} ||x_n - y_m||^2, 0 < \kappa \ll 1$
- Repeat until convergence:
  - E-step: Compute $\mathbf{P}$,
    $\mathbf{P}_{nm} = \frac{\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_m - t||^2)\exp(\kappa\vec{x}_n^\top R\vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_k - t||^2)\exp(\kappa\vec{x}_n^\top R\vec{y}_k)) + \frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}}$
  - M-step:Solve for $\tilde{R}, s, t, \sigma^2$ and κ
    - $N_P = \mathbf{1}^\top\mathbf{P}\mathbf{1}$
    - $\mu_x = \frac{1}{N_p}\mathbf{X}^\top\mathbf{P}^\top\mathbf{1}$, $\mu_y = \frac{1}{N_p}\mathbf{Y}^\top\mathbf{P}^\top\mathbf{1}$
    - $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{1}\mu_x^\top, \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\mu_y^\top$
    - Minimize
      $Q(s, \tilde{R}, \sigma^2, \kappa) = \frac{1}{2\sigma^2}(\mathrm{tr}(\hat{\mathbf{X}}^\top \mathrm{diag}(\mathbf{P}^\top\mathbf{1})\hat{\mathbf{X}}) - 2s \cdot \mathrm{tr}((\hat{\mathbf{X}}^\top\mathbf{P}^\top\hat{\mathbf{Y}})^\top\tilde{R}) + s^2\,\mathrm{tr}(\hat{\mathbf{Y}}^\top\mathrm{diag}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}})) - \kappa(\mathrm{tr}((\vec{\mathbf{X}}_{2d}^\top\mathbf{P}^\top\vec{\mathbf{Y}}_{2d})^\top\tilde{R}) + \vec{\mathbf{X}}_z^\top\mathbf{P}^\top\vec{\mathbf{Y}}_z) + N_P\log\sigma^2 - N_P\log(\frac{\kappa}{2\pi(e^\kappa - e^{-\kappa})})$
    - Use numerical optimization library and derivatives in Table A.1.
    - $t = \mu_x - s\tilde{R}\mu_y$
- Compute orientations: $\vec{\mathbf{Y}}^{final} = \vec{\mathbf{Y}}R^\top$.
- Compute coordinates: $\mathbf{Y}^{final} = s\mathbf{Y}\tilde{R}^\top + \mathbf{1}t^\top$.

Box 4.2. Algorithm for computing a similarity transform using both coordinate and orientation of a line.

4.4.2.3 Algorithm 3: Elastic transformation

If the data is distorted, which is often the case for electron tomograms, a similarity transform as computed by algorithm 2 might not be sufficient to obtain a satisfying result. In this case, an elastic registration of the ends is needed. Myronenko and Song [17] formulated an elastic deformation in this framework as individual translation vectors $\nu(y)$ for each $y \in Y$ so that

$$T(Y, \nu) = Y + \nu(Y). \tag{4.41}$$

Coupling the transformation of coordinates and directions is hard because a rotation that would apply to orientations is not explicitly formulated for coordinates by the displacement vectors ν in Equation 4.41. In practice, however, we ignore changes in the orientation because we can compute a rigid alignment prior to elastic registration and therefore know that the remaining transformation contains negligible rotation.

M-step With this assumption we can rewrite Equation 4.16 with the orientation as

$$f(\nu, \sigma^2, \kappa) = Q_c(\nu, \sigma^2) + \frac{\lambda}{2}\Phi(\nu) + Q_d(\kappa), \text{ with} \tag{4.42}$$

$$Q_d(\kappa) = -\kappa\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)\vec{x}_n^\top\vec{y}_m - N_P\frac{\kappa}{e^\kappa - e^{-\kappa}} + \text{const.}, \text{ and} \tag{4.43}$$

$$Q_c(\sigma^2) = \frac{1}{2\sigma^2}\sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m|x_n, \vec{x}_n)||x_n - (y_m - \nu(y_m))||^2 + N_P\log\sigma^2 + \text{const.} \tag{4.44}$$

Because Q_d is independent of the displacement vectors μ, the optimization can be performed exactly as described in Myronenko and Song [17]. The only difference here is, that the prior $P(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}})$ depends on κ, which must also be updated at each M-step. κ can be updated in each iteration again with Newton's method like in Section 4.4.2.2. With this strategy, the orientation only influences the result via the posterior $p^{old}(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}})$.

E-step The posterior again is computed by Bayes' rule

$$P(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}}) = \frac{P(\mathbf{x}, \vec{\mathbf{x}}, \mathbf{m}, \vec{\mathbf{m}})}{P(\mathbf{x}, \vec{\mathbf{x}})} = \frac{\exp(-\frac{1}{2\sigma^2}||x - (y_m + (\mathbf{G}(m,\cdot)\mathbf{W})^\top)||^2)\exp(\kappa\vec{x}^\top\vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n - (y_k + (\mathbf{G}(m,\cdot)\mathbf{W})^\top)||^2)\exp(\kappa\vec{x}_n^\top\vec{y}_k)) + \frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}} \tag{4.45}$$

The algorithm is summarized in Box 4.3.

Non-linear line end registration:

- Initialization: $\mathbf{W} = 0, \sigma^2 = \frac{1}{NM}\sum_{m,n=1}^{M,N}||x_n - y_n||^2, 0 < \kappa \ll 1$
- Construct $\mathbf{G}$: $g_{ij} = \exp(-\frac{1}{2\beta^2}||y_i - y_j||^2)$ and set $0 \leq w \leq 1, \beta > 0, \lambda > 0$
- Repeat until convergence:
  - E-step: Compute $\mathbf{P}$,
    $\mathbf{P}_{mn} = \frac{-\exp(\frac{1}{2\sigma^2}||x_n-(y_m+(\mathbf{G}(m,\cdot)\mathbf{W})^\top)||^2)\exp(\kappa\vec{x}_n^\top\vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n-(y_k+(\mathbf{G}(m,\cdot)\mathbf{W})^\top)||^2)\exp(\kappa\vec{x}_n^\top\vec{y}_k))+\frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}}$
  - M-step:
    - Solve $(\mathbf{G} + \lambda\sigma^2\operatorname{diag}(\mathbf{P1})^{-1})\mathbf{W} = \operatorname{diag}(\mathbf{P1})^{-1}\mathbf{PX} - \mathbf{Y}$ for $\mathbf{W}$,
    - $N_P = \mathbf{1}\mathbf{P}^\top\mathbf{1}$, $\mathbf{T} = \mathbf{Y} + \mathbf{GW}$,
    - $\sigma^2 = \frac{1}{2N_P}(\operatorname{tr}(\mathbf{X}^\top\operatorname{diag}(\mathbf{P}^\top\mathbf{1})\mathbf{X}) - 2\operatorname{tr}((\mathbf{PX})^\top\mathbf{T}) + \operatorname{tr}(\mathbf{T}^\top\operatorname{diag}(\mathbf{P1})\mathbf{T}))$
    - iterate $\kappa = \kappa - \frac{-\operatorname{tr}((\vec{\mathbf{X}}^\top\mathbf{P}^\top\vec{\mathbf{Y}})^\top)-N_P(\frac{1}{\kappa}-\coth\kappa)}{-N_P(-\frac{1}{\kappa^2}+\frac{1}{\sinh^2\kappa})}$, until convergence
- Compute coordinates: $\mathbf{Y}^{final} = \mathbf{Y} + \mathbf{GW}$

Box 4.3. Non linear registration algorithm. The only difference to Myronenko and Song [17], Figure 4 therein, is the computation of the posterior $p^{old}(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}})$ and update of κ.

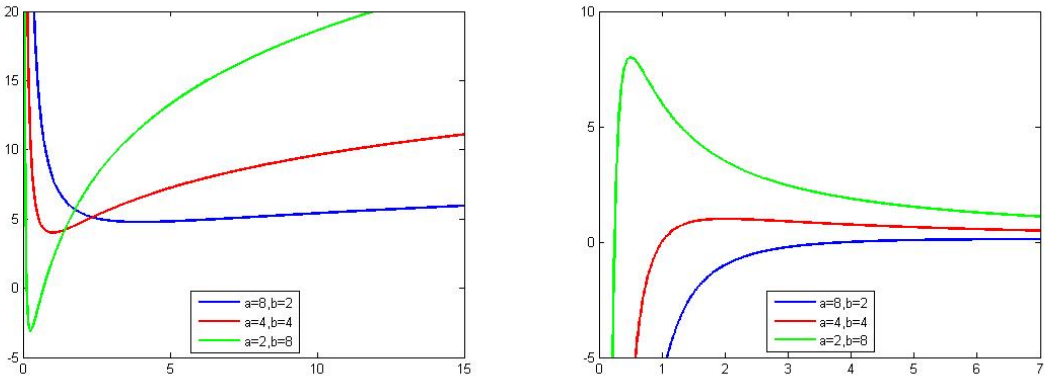
4.4.2.4 Implementation

To optimize Q for the similarity transform (Section 4.4.2.2), we use IPOpt [167], a free optimization library. As starting point for the optimization, we use the parameters from the previous step, except for σ^2. In our experiments, σ^2 is not optimized correctly if the initial value is placed on the wrong side of the peak of the derivative of Q. The derivative of Q with respect to σ^2 is shown in Figure 4.11(b). To avoid this, we initialize σ^2 as

$$\sigma^2 = \frac{1}{2N_P}(\operatorname{tr}(\hat{\mathbf{X}}^\top\operatorname{diag}(\mathbf{P}^\top\mathbf{1})\hat{\mathbf{X}}) - 2s\operatorname{tr}((\hat{\mathbf{X}}^\top\mathbf{P}^\top\hat{\mathbf{Y}})^\top\tilde{R}) + s^2\operatorname{tr}(\hat{\mathbf{Y}}^\top\operatorname{diag}(\mathbf{P1})\hat{\mathbf{Y}})),$$

which would be the minimum of σ^2 if the other parameters were fixed. In our experiments this was sufficient to guarantee that the initial σ^2 is placed on the right side.

Furthermore, the smaller σ becomes, the sharper the peak gets that is the minimum of the objective function. This is shown in Figure 4.11(a). This sharp peak might make it hard to estimate a reasonable step size when optimization is done numerically. We believe that this is the reason, why IPOpt can not always find an optimum in later expectation maximization iterations in our implementation. However, for our test data, we found that about 4000 of IPOpt iterations are sufficient to find a new parameter set that decreases the log-likelihood.



(a) (b)

Figure 4.11. (a) shows the objective function $Q = a/\sigma^2 + b \log \sigma^2$ (Equation 4.40) plotted over σ^2 for different values of the terms a and b. The smaller the ratio a/b, the sharper the minimum becomes. (a) shows the derivative $\partial Q/\partial \sigma^2 = -a/\sigma^4 + b/\sigma^2$ plotted over σ^2 for different values of the terms a and b. If σ^2 is initialized right of the peak, the minimum might not be found when optimizing numerically.

We use the moving least squares algorithm as described by Schaefer et al. [145] to apply the deformation computed by the elastic registration method (Box 4.3) to the full lines as well. To do so, we use the displacements vectors $\mathbf{G}(m, \cdot)\mathbf{W}$ as landmarks for moving least squares.

The running time for Algorithm 3 is $O(M^3)$ but can be reduced to linear by computing P with a fast Gauss transform and G by a low rank approximation as described in [17]. We did not implement this technique, which is why our implementation has cubic running time.

4.4.3 Final assignment of endpoints

The previously described method estimates probability distributions to individual assignments. However, joint assignments of several lines are not formulated and conflicting assignments are not considered. To allow modeling beliefs of joint assignments of line ends, we use the probabilistic graphical model formulation as discussed in Section 4.3.5. We first describe how we fill factors in Section 4.4.3.1 by making use of line orientation. We find the MAP assignment with belief propagation which not always converges for our data. Section 4.4.3.2 describes how well-directed user input can boost the performance of belief propagation in terms of convergence of this algorithm significantly. Because the method requires parameters, we explain in Section 4.4.3.3 how we obtain a good parameter set for true biological data.

4.4.3.1 Factor values for matching microtubule end points

We use three types of information to fill the factors Φ_o in Equation 4.17. First of all, we assume microtubules are rather straight and therefore expect the angle $\mathrm{d}_\alpha(x, y) = \mathrm{acos}(\vec{x} \cdot -\vec{y})$ to be close to 0. Second, we assume that the coordinates of matching line ends are close and therefore compute the distance $\mathrm{d_c} = ||x - y||$. $\mathrm{d_c}$ measures the distances on a plane, which we assume all lines end on, as illustrated in Figure 4.12. Third, we compute the *projected distance* $\mathrm{d_p}$ which we define as the distance of x to the intersection of the line going through y and a plane that is defined by x and its normal $\vec{x}$. Figure 4.12 shows a sketch of d_p. We write $\mathrm{d_p}(x, y)$, $\mathrm{d_c}(x, y)$ and $\mathrm{d}_\alpha(x, y)$ for the projected distance, coordinate distance and angle of x and y respectively.

We compute entries for Φ_o from these three values $\mathrm{d}_\alpha, \mathrm{d_c}$ and $, \mathrm{d_p}$ as follows: We define

$$\Phi_o(\mathbf{x}_i = y_j) = \Phi_p(\mathbf{x}_i = y_j)\Phi_\alpha(\mathbf{x}_i = y_j)\Phi_c(\mathbf{x}_i = y_j), \tag{4.46}$$

and fill the three factors by exponential distributions with parameters λ_p, λ_c, λ_α, for example, $\Phi_p(\mathbf{x}_i = y_i) = \lambda_p \exp(-\lambda_p \mathrm{d_p}(xi, y_j))$ (see also Amat et al. [86]).

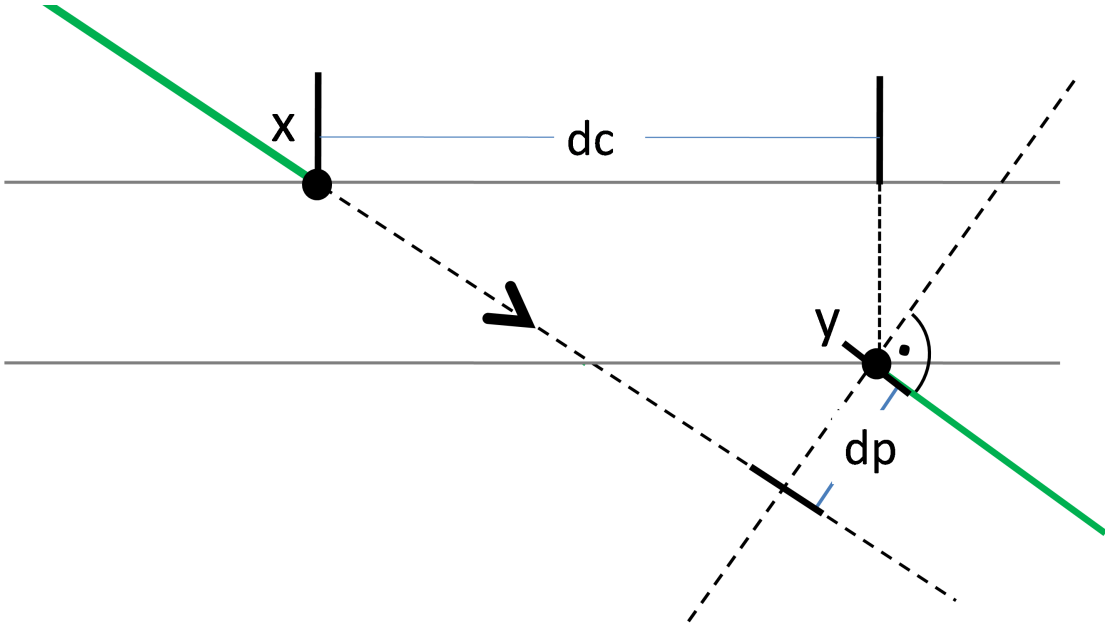


Figure 4.12. Illustration of coordinate distance $\mathrm{d_c}(x, y)$ and projected distance $\mathrm{d_p}(x, y)$.

To account for joint assignments of pairs, we fill the pair factors as in Amat et al. [86] (see Section 4.3.5.1) by

$$\Phi_{oo}(\mathbf{x}_i = y_k, \mathbf{x}_j = y_l) = \lambda_s \exp(-\lambda_s \, \mathrm{d_s}(x_i, y_k, x_j, y_l)), \tag{4.47}$$

but in contrast to Amat et al. we set the diagonal to 0 to account for the mutual exclusiveness constraint (Koller and Friedman [162]).

To reduce the number of factors, we restrict the valid assignments of a node $x \in X$ to close nodes $y \in Y$ where close means $\mathrm{d_c}(x, y) < t_c$, $\mathrm{d_p}(x, y) < t_p$ and $\mathrm{d_\alpha}(x, y) < t_\alpha$. We then omit all pair factors for variables that cannot be assigned to the same y_i. In practice, this causes the distribution to decompose into several independent joint distributions of subsets of the random variables that can be solved independently.

4.4.3.2 User-given evidence to ensure convergence of loopy belief propagation

We use belief propagation to infer the MAP assignment from the distribution. Because convergence is not guaranteed (Section 4.3.5.2), we identify those nodes which cause oscillation. We then point an expert to these nodes and let them decide on an assignment. The opinion on a particular assignment is incorporated into the statistical model by setting the chosen assignment in all associated factors to 1 and all other entries to 0. Belief propagation is then run again to see if all conflicts were solved. The two steps, manual node assignment and belief propagation, are iterated, until belief propagation converges.

To identify nodes that cause oscillation, we compute disagreement of two messages μ_i, μ_j about variable $\mathbf{x}_k$ as the L^∞ norm of the difference vector, $||\mu_i - \mu_j||^\infty = \max(\mu_i(\mathbf{x}_k = y_1) - \mu_j(\mathbf{x}_k = y_1), ..., \mu_i(\mathbf{x}_k = y_l) - \mu_j(\mathbf{x}_k = y_l))$, where l is the number of states of $\mathbf{x}_k$. We collect for each variable each incoming messages on all factors involved with the variable. For example, in Figure 4.9, we would collect messages 1,2 and 3 for variable $\mathbf{x}_1$, messages 4 and 6 for variable $\mathbf{x}_2$ and so on. We compute for each pair of messages the disagreement. We define the maximum disagreement for each variable as the maximum of the computed differences. For each independent network in the factor graph, the variable that messages disagree on most is then shown to the expert. We refer to these nodes as *critical* nodes.

4.4.3.3 Parameters

The PGM algorithm for point matching has 11 parameters: the four weights for computing the factor entries ($\lambda_p, \lambda_d, \lambda_\alpha, \lambda_s$), the three threshold parameters t_p, t_d, t_α, and four values that correspond to an assignment to the dummy node y_0. We denote them by d_p^0, d_d^0, d_α^0, and d_s^0. We compute $d_x^0 = -\log(r)/\lambda_x$ where r can be chosen in the range of $[0, 1]$. The dummy values are thereby chosen at the point where the cumulative distribution function of the exponential distribution $1 - e^{-\lambda d^0}$ reaches $1 - r$. The thresholds t_p, t_d, t_α are set to d_p^0, d_d^0, d_α^0 respectively.

We determine good parameters for the weights $\lambda_p, \lambda_d, \lambda_\alpha, \lambda_s$ with a maximum-likelihood estimate using the provided ground truth. To do so, we consider Eq. 4.17 as being the likelihood function for the distribution when only one sample was seen. Because the maximum-likelihood estimate for the parameter of an exponential distribution is just the mean of the data, we can compute $\lambda_x = \frac{1}{N} \sum_{i=1}^{N} \mathrm{d_x}(x, y)$. We will see in the result section how the final choice is made. For example, Figure 4.19(a) shows a histogram of d_α measured for all pairs in the ground truth. The red curve shows the maximum-likelihood estimate of the exponential distribution. λ_α and t_α, d_α^0 are chosen as indicated by the blue and the green line, respectively. We will discuss this figure more in the result section.

The parameter choice influences two aspects of the result. First of all, while manual input about assignments necessarily enforces convergence of belief propagation after a finite number of steps, we cannot say in advance 1) how many nodes must be assigned manually to yield convergence and 2) how many iterations are needed. To measure this, we experimentally test the procedure on true biological data. We use the ground truth created by the expert. We iteratively

- Run belief propagation (5 passes), and
- Assign the nodes that cause oscillation automatically by reading the assignments from the ground truth,

until belief propagation converges. The number of iterations and the number of needed assignments for this procedure on typical data are given in Section 4.6.3.

Second, the parameter choice might influence the quality of the result. We can measure three types or errors: 1) False negatives (FN): A pair assignment in the ground truth is missing in the automatically computed matching. 2) False positives (FP): A pair was computed that is not at all in the ground truth. 3) Disagreements (D): A node was matched differently in ground truth and automatic assignment. Furthermore, we can measure how many pairs were assigned correctly, that is, the true positives (TP). To estimate the sensitivity of the parameters $\lambda_p, \lambda_d, \lambda_\alpha, \lambda_s$, we vary their value around the maximum-likelihood estimate. For each value, we run belief propagation and assure convergence using the ground truths as described before. We measure the number of iterations, the number of needed assignments, and the number of FN, FP and D resulting from the computed matching. Results for this procedure are given in Section 4.6.3.

Furthermore, to get an estimate on the error rate, we will provide the precision P=$\frac{TP}{N_A}$, recall R=$\frac{TP}{N_M}$, and fraction of disagreeing matchings DIS=$\frac{D}{N_M}$for true biological data. N_M here is the number of pairs in the ground truth and N_A the number of pairs in the automatic matching.

4.4.3.4 Implementation

We use the libDai library [168] to run belief propagation on the network where we like in Amata et al. use the scheduling proposed by Elidan et al. [169]. The graphical user interface was implemented in Amira [127].

4.5 Data

4.5.1 Sample preparation, imaging and tomogram reconstruction

Samples were prepared and imaged at MPI-CBG Dresden, EMBL Heidelberg and University of Colorado at Boulder by our collaborators as described in the following paragraphs. We repeat here the protocol description from our previous publication [170] as a reference for the interested biologist.

Sample preparation Three different types of samples were used. Samples were prepared as previously reported (Müller-Reichert et al. [122], Höög et al. [171], Hannak and Heald [172] and Murray [173]). In brief: 1) Bipolar spindles were assembled from *X. laevis* egg extract. 2) Single cell *C. elegans* embryo expressing GFP-tagged tubulin were sucked into capillary tubes at metaphase and 3) *T. brucei* cells were grown at logarithmic phase. The specimens were high-pressure frozen using a Leica EMPACT2 or HPM-010 high-pressure freezer, freeze-substituted in a Leica AFS2. The substitution cocktail contained 1% osmium tetroxide and 0.1% uranyl acetate in acetone for *T. brucei* and *C. elegans* and 0.1% Tannic Acid, 0.2% glutaraldehyde, 2.5% water for *X. laevis*. Specimens were thawed, infiltrated and embedded in epoxy (Epon) or metacrylate (HM20) resins. Semi-thin (300-350 nm thickness) serial sections were cut, collected on forvar-coated slot copper grids for EM. *C. elegans* and *T. brucei* samples were post-stained in 2% uranyl acetate in 70% methanol (15 min) followed by lead citrate (5 min). *X. laevis* sections were post-stained only with lead citrate.

Electron tomography For electron tomography, 10 nm colloidal gold fiducial particles were deposited on the grids. Electron tomography tilt series were acquired in a Tecnai F30 electron microscope (FEI Company Ltd., Eindhoven, The Netherlands) operated at 300 kV at 1° increments between -60 and +60 degrees, with the SerialEM software (Mastronarde [124]) using a Gatan US1000 2k camera or a FEI 4k Eagle camera. The pixel size was ~2 nm. The tomograms were reconstructed and flattened using he IMOD software package as described in Kremer et al. [96]. Due to the large area of interest, 3×3 montages (11.4×11.4 μm) were acquired 3 or 4 times per section for *X. laevis*, before being joined into large supermontages 11.4×32 μm).

Microtubule tracing Microtubule centerlines in tomograms of *X. laevis* and *C. elegans* samples were traced automatically as described by Weber et al. [100]. The supermontage data was binned prior to calculation of microtubule centerlines. Automatically traced segments were manually validated and corrected using Amira [127].

The segmentation for *T. brucei* was performed manually by an expert using the IMOD software (Kremer et al. [96]).

4.5.2 Evaluation data

Datasets We chose three pairs of consecutive sections from centrosomes of *C. elegans* mitotic spindles to validate our approach. Line ends are spread in a ring-like structure around the microtubule organizing center as depicted in Figure 4.14(a). The orientation of the lines is depicted in Figure 4.14(b). The example data sets contain about 1500 lines per section.

We furthermore use a stack of three sections of microtubules within the *X. laevis* meiotic spindles. This data is particularly challenging, since it is badly deformed, making it hard to find a proper transformation. Furthermore, microtubules run in dense bundles (see Figure 4.5(c)) which makes line end matching even more challenging. Figure 4.14(d) and 4.14(c) show orientation and coordinates of line ends of a small area of these sections. Here, the orientation is clustered, because unlike in the centrosome data, microtubules have a similar orientation. Each section contains about 3500 lines.

Third, we show results for the sub-pellicular microtubule skeleton of *T. brucei*, an array of parallel microtubules underlying the plasma membrane (reviewed in Gull [23], Farr and Gull [22]). Figures 4.14(f) and 4.14(f) show again coordinates and angle of line ends of two consecutive sections of our test dataset. Each section contains about 200-400 lines.

4.5.3 Generating a ground truth

To prepare a ground truth for the chosen *C. elegans* and *X. laevis* data, we first created a matching using the described pipeline and then let an expert correct the computed matchings. Because the final PGM point matching (Section 4.4.3) needs manual input, the programmer assigned evidence as she saw fit to obtain a solution. This solution was then passed on to the biological experts for further corrections.

To prepare the ground truth for *C. elegans* data, we computed an alignment and the line end correspondences for three section pairs using our proposed pipeline. We did not perform the elastic registration described in Section 4.4.2.3. Two experts then corrected the computed matching by first looking for ends that were wrongly connected and correcting these and then for missing connections among all the unconnected ends.

For *X. laevis*, we generated a ground truth for three consecutive sections. The lines were first aligned using the ICP algorithm. We did not use the elastic deformation algorithm described in Section 4.4.2.1 and Section 4.4.2.3 to align the endpoints further since the method was not available when generating the ground truth. Since these sections are huge, we could not let an expert correct all matchings. We cut out a small part of the sections that contained about 1000 lines per section. The matchings in this subset of lines, computed as described in Section 4.4.3, were then corrected by the expert.

We created no ground truth for the *T. brucei* data.

4.6 Results

4.6.1 Initial transformation

We tested the initial alignment on more than 50 pairs of consecutive line sets from *C. elegans* samples (Section 2.3). In about 10% of the cases, we had to scale one line set by $\pm 5\%$ to obtain a reasonable result. The algorithm then always found a reasonable initial alignment. We used 50 line ends in each section to build the DCG.

Furthermore, we tested the initial alignment on four subsequent line sets of *X. laevis* samples where each section contained about 3500 lines. Here, we had to pick at least 100 nodes from each section, restrict the size of the cliques in the DCG to 10 nodes and test different scalings to get a reasonable result.

We tested the algorithm on the *T. brucei* dataset as depicted in Figure 4.13(a). Here, we rotated each section so that the rotation angle between consecutive sections is about 180°. The algorithm failed to reliably compute an initial transformation. For this sample, no matching points could be identified by comparing angles of line orientations because all lines run more or less in parallel (Figure 4.13(b)).

4.6.2 Registration

Robustness Furthermore, we tested performance of the rotation estimation (Box 4.1) and the similarity transform computation (Box 4.2). We rotated four datasets over 360° in 5° steps and measured whether the registration algorithm was able to discover the true rotation angle using only coordinates (see Myronenko and Song [17]), only orientation (Box 4.1) or both coordinates and orientation (Box 4.2) of the line ends as features. We chose two sets from the *C. elegans* ground truth and two randomly chosen examples from parts of *X. laevis* sections. We examined if the true rotation angle could be computed and furthermore measured the final κ and σ^2.

Results for the computed rotation angle with each algorithm is in Table 4.4. The mean and standard deviation (std) for σ^2 was 8.68e-005 and 1.25e-004 when the rotation angle was correctly computed (white in Table 4.4) and 0.0692 and 0.0709 when the angle was not computed correctly (black in Table 4.4). Mean and std of κ where 123 and 16.4 for correct and 22.7 and 18.9 for incorrect results. Correct results can be clearly distinguished from incorrect results by inspecting κ and σ^2.

We also tested the algorithms on the *T. brucei* dataset (Figure 4.13(a)). The original algorithm for computing a similarity transform (see [17]) and our modification taking into account both coordinates and orientations (Box 4.2) both failed to compute a correct result for the unaligned stack (Figure 4.13(a)). However, using line orientations alone (Box 4.1), the true rotation could be computed (Figure 4.13(c)). With this initialization, we could compute a similarity transform of the ends with the algorithm in Box 4.2 (Figure 4.13(d)).

Computed displacements for elastic registration We compared the distances of nodes after elastic registration computed by the original CPD algorithm [17] and our modification to the

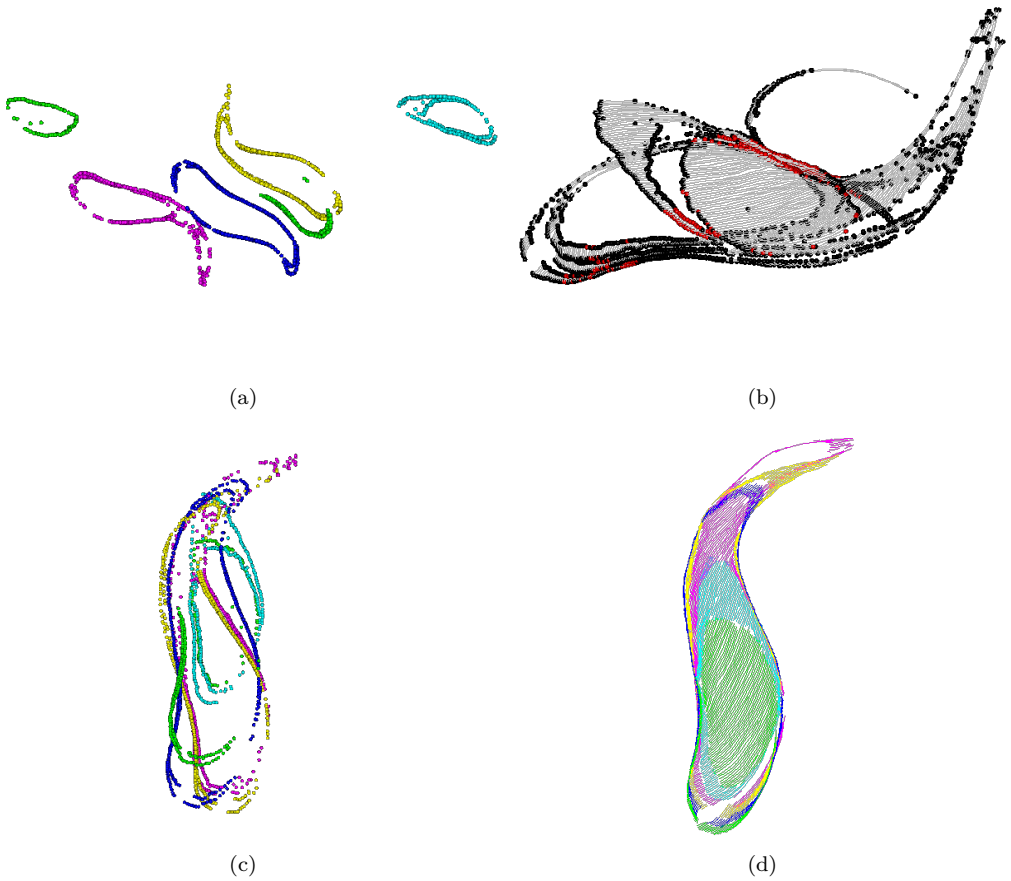


Figure 4.13. (a) The *T. brucei* dataset before aligning. (b) The *T. brucei* dataset after running the initial alignment as described in Section 4.4.1. Red: Nodes that were picked for computing the transformation. The uniform orientation of and spacing between lines makes it very hard to decide on potential pairs. In this example, the algorithm failed to compute a reasonable transformation. (c) shows the line ends of the sample after computing rotation using the algorithm in Box 4.1. The end coordinate alignment is not accurate because it was estimated based on the posterior that was computed from the orientations only. (d) shows the result of the similarity transform computation as described in Box 4.2 (w=0.9).

original CPD algorithm (Box 4.3). We measured the distance of the manually connected ends before and after the registration on the *X. laevis* ground truth. Figure 4.15 shows the histograms of the distances. Both methods perform equally well. However, without taking into account line orientation, CPD needed roughly 1.5 times as many iterations. We measured the same factor for other datasets we tested.

Figure 4.16 shows a polar plot of displacement vectors that were computed for two full *X. laevis* sections with the elastic registration method described in Section 4.4.2.3. The sections were aligned with a similarity transform as described in Section 4.4.2.2 before computing the elastic registration.

Limitations The elastic registration for the *T. brucei* dataset yielded a reasonable result if the data was pre-aligned as just described. However, some parts were not aligned properly, see Figure 4.17(a) and 4.17(b). Here, the segmented lines were often too long or too short. This violates the prior assumption that point displacements are coherent in a small neighborhood. The proposed model for matching cannot deal with these cases.

When imaging with an electron microscope, the position of the relevant part of a sample is often hard to find and subsequent tomograms might thus be shifted. In this case, only a small portion of the lines in one section has a matching counterpart in the next section. To test, if the

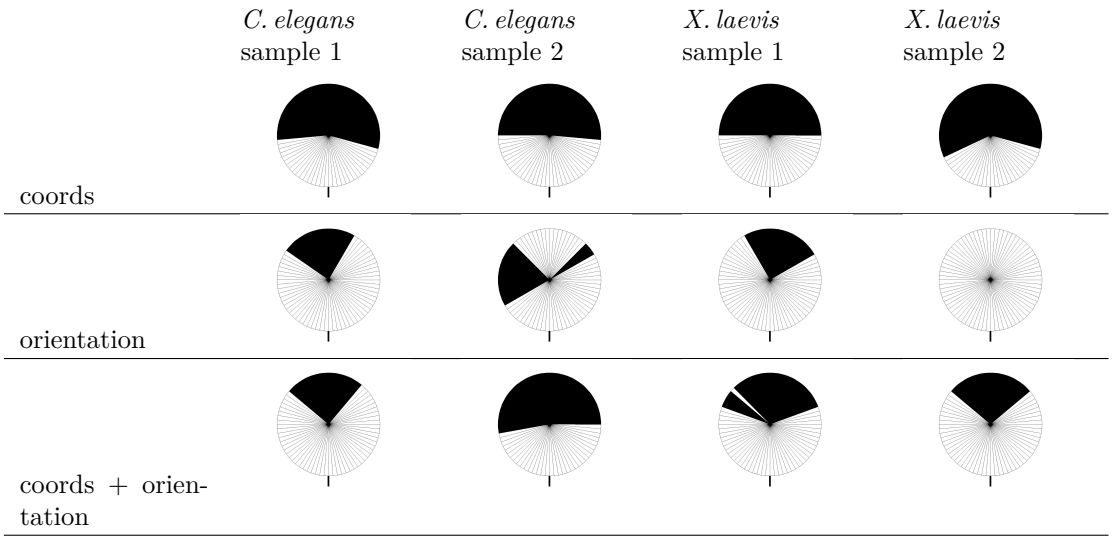


Table 4.4. Result for the similarity transform alignment as described in Myronenko and Song [17] (first row), orientation alignment Box 4.1 (second row) and similarity transform Box 4.2(third row). We tested 360° rotation at a 5° step size as depicted by the colored circles. Black marks a failure of the algorithm to recover the true rotation angle from the artificially rotated dataset and white marks a correct result. 0° angle is indicated by a vertical bar.

algorithm can deal with this, we cut out two regions with little overlap from an *X. laevis* dataset (Figure 4.18(a)) and run the registration algorithms on it. No version of the alignment algorithms yielded a reasonable result (Figure 4.18(b) and (c)).

4.6.3 Point matching

Parameters for the probabilistic graphical model matching We used the ground truth of the *X. laevis* sample to estimate parameters for the probabilistic point matching with a maximum-likelihood fit as described in Section 4.4.3.2. Before elastic point registration, parameters were estimated as 771, 577, 297, 5.8 for $\lambda_c, \lambda_p, \lambda_s$ and λ_α. After registering points using the modified CPD (Box 4.3) parameters were estimated as 243, 170, 293 and 5.8 for $\lambda_c, \lambda_p, \lambda_s$ and λ_α (see Figure 4.19 and 4.15).

To fine tune the parameters and test their sensitivity, we varied each parameter separately about its maximum-likelihood estimate. For each parameter set, we measured the performance of the PGM matching as described in Section 4.4.3.3. Figure 4.20 shows the resulting curves for the registered dataset. Here, we give absolute values for number of false negatives and disagreements instead of ratios. We chose the pair shift parameter as 150 because this seems to yield better results in terms of number of iterations the user has to perform (see Figure 4.20(b)) and left all other parameters at their maximum-likelihood estimates for further experiments.

Quality of matching We evaluated the final result of the PGM approach for line matching on the ground truth created for three sections of an *X. laevis* spindle (see Section 3.5.2). We set the parameters to the values obtained in the previous section. Table 4.5 (configuration C1) shows the results for end point assignments using a PGM or computing a MWM.

To examine the effect of a variation of the full parameter set on the quality of the result, we ran the algorithm again, setting all parameters to twice their original value. Results for both PGM matching and MWM with this parameter set is given in Table 4.5 (configuration C2). To test stability of the PGM with respect to noise, we tried the matching on the *X. laevis* ground truth, before registering the endpoints. We used maximum-likelihood parameters as estimated before: $\lambda_c = 771, \lambda_p = 577$ and $\lambda_\alpha = 5.8$ except for the pairwise shift λ_s, which we again set to 150. The result of the PGM approach is unaffected by parameter variations and noise in terms of precision (about 0.96), recall (about 0.95) and fraction of disagreeing matches (0.04). MWM performs worse

Results for the *X. laevis* data

sample	config	P (PGM)	P (MWM)	R (PGM)	R (MWM)	DIS (PGM)	DIS (MWM)	$\frac{\#assignments}{N_M}$	#assignments
X. laevis	C1	0,956	0,901	0,951	0,904	0,038	0,088	0.009	4
X. laevis	C2	0,950	0,851	0,953	0,874	0,039	0,121	0.026	8
X. laevis	C3	0,956	0,609	0,948	0,634	0,038	0,349	0.028	19

Results for the *C. elegans* data

sample	expert	P (PGM)	P (MWM)	R (PGM)	R (MWM)	DIS (PGM)	DIS (MWM)	#iterations	#assignments
C. elegans 3 samples	two experts	0,946	0,934	0,975	0,97	0,01	0,018	0	0

Table 4.5. Comparison of results for the probabilistic matching (PGM) and a maximum weighted matching (MWM). C1: Point matching results after CPD registration with parameters $\lambda_c = 243$, $\lambda_p = 170$, $\lambda_\alpha = 5.8$ and $\lambda_s = 150$. C2: Point matching results after CPD registration with parameters $\lambda_c = 486$, $\lambda_p = 340$, $\lambda_\alpha = 11.6$ and $\lambda_s = 150$. C3: Point matching results before CPD registration with parameters $\lambda_c = 771$, $\lambda_p = 577$, $\lambda_\alpha = 5.8$ and $\lambda_s = 150$. Results for the *C. elegans* data show the mean measured on three samples where a ground truth was provided by two experts. Parameters for the PGM matching were $\lambda_c = 243$, $\lambda_p = 170$, $\lambda_\alpha = 5.8$ and $\lambda_s = 150$. P=$\frac{TP}{N_A}$measures the fraction of pairs found by an automatic method were correct. R=$\frac{TP}{N_M}$measures the fraction of pairs in the ground truth that were correctly computed by the automatic method. DIS=$\frac{D}{N_M}$is the fraction of pairs in the automatic matching, that disagree with the assignment in the ground truth.

with precision and recall dropping below 0.9 when parameters are changed and below 0.65 with the unregistered lines. In all cases, only a small fraction of nodes had to be assigned manually to enforce convergence of loopy belief propagation ($< 3\%$).

It took an expert about 1 h to assign 100 critical nodes detected by the PGM approach manually. Checking unconnected lines after the final matching was computed took about 1h for 500 ends.

To measure the quality of the stitching for *C. elegans* stacks, we compared results of the PGM and MWM matching on the six ground truths (3 datasets, 2 experts) described in Section 4.5.2. The lower part of Table 4.5 shows the mean of the performance of PGM and MWM. Both approaches perform equally well on the data (precision and recall 0.97 and 0.96 respectively) and we did not measure a significant difference between experts. For the PGM all networks always converged without need of user input.

Limitations We tested the PGM point matching on the *T. brucei* dataset after elastic registration. We found that it is difficult to assign evidence here. Figure 4.17(c) and 4.17(d) show examples of points that the algorithm is requesting the user to assign as described in Section 4.4.3.2. Here, we could not decide if lines were missing (4.17(c)) and which nodes correspond (4.17(d)). Because the PGM point matching relies on user input, this method therefore cannot be applied to stitch this data.

4.7 Discussion

In this chapter we have presented an automated solution for joining of filaments that span two or more serial sections of a sample. We tested our software on three true biological samples with possible errors, imprecision and noise in the tracings. Results show an encouraging low number of wrong and missing connections (less than 5% for both). We believe that the success of the software results from our implementation of line orientation as an additional feature for registration and matching, as well as the usage of probabilistic methods that naturally deal with noise and outliers.

To the best of our knowledge, the usage of additional information on orientation for stitching has yet received little attention. We exploit this property in each step of our pipeline (initial transform, registration and final end point matching). The pipeline is similar to that used when aligning serial sections of neurons imaged with confocal microscopy. Hogrebe et al. [159] and Dercksen et al. [132] use segmented neuron centerlines to compute the registration of the image data. The major difference in our application is that we a) include information about the line orientation in all three steps, and b) use probabilistic methods to compute results. The latter is essential in our case, because lines extracted from electron tomograms are subject to noise and artifacts and the uncertainty of results needs to be modeled to achieve reliable results.

Taking into account orientation for the registration (Section 4.4.2, Box 4.1 and Box 4.2) has several advantages. First of all, orientation stabilizes the result if subsequent sections are subjected to large rotations (Table 4.4 and results for the *T. brucei* dataset). Taking into account orientation helps avoiding local maxima that would arise if the lines were aligned only considering coordinates. This is expected if the orientation clusters, as in *X. laevis* and *T. brucei* (Figure 4.14(d) and Figure 4.14(f)), but also helps if the orientation just forms a different pattern than the coordinates do as in *C. elegans* (Figure 4.14(b)). Second, taking orientation into account for the elastic registration speeds up convergence of expectation maximization by a factor of two in our experiments. Finally, considering line orientation allows an estimation of the quality of the result by inspecting the concentration parameter κ after alignment in addition to inspecting σ^2. This ability to assess a result without having a ground truth available is in general a huge advantage of probabilistic approaches.

The numerical optimization step for Algorithm 2 (Box 4.2) is an unfortunate consequence of the scaling parameter s. Without the scaling, the objective function Q could probably be minimized by formulating the transformation as dual quaternions and solving the transformation parameters as described by Walker [139]. While this still would not provide a final optimization in closed form, it might be possible to calculate closed form update equations for a generalized expectation maximization. Note that we cannot make use of other approaches that use scaled vectors such as Wells [154] because our orientations are unit vectors.

The probabilistic approach for final point matching (PGM) clearly outperforms results obtained by a maximum weighted matching (MWM) if the data is noisy (Table 4.5). Furthermore, the results are not very sensitive to parameter variations (Table 4.5 and Figure 4.20). This is an important aspect of the algorithm, since new data will require an adjustment of the parameters. We believe that one reason for the good performance of PGM is that the pair factors enforce a coherent shift of neighboring assignments. Another reason is, that we incorporated user input into the mathematical model. Manually assigned nodes are not only correct in the final result, but also influence the decision on neighboring connections of line ends when using the PGM approach we described. Assigning the nodes manually is tedious, but in all cases, less than 3% of the line ends had to be connected manually, which is an acceptable effort considering the quality of the final result.

There is no great difference in the results between MWM and PGM matching if the data is pre-aligned well and the correct parameters are adjusted (Table 4.5, *X. laevis* C1 and *C. elegans*). For the *C. elegans* data, this is mainly due to the fact that orientation of lines narrows down possible candidates a lot (see Figure 4.3(a) for an example) and there are few ambiguities the algorithm has to decide on. However, if parameters are not exactly known, and the data is noisy, the PGM approach is a much safer choice for point matching.

Despite these encouraging results, care must be taken, when analyzing the resulting 3D line sets. Errors in the stitching will results in a biased statistics on length and number of microtubules. From Table 4.5, we have seen that ~95% of the correct line connections were found and $\sim 4\%$ (*X. laevis*) and $\sim 1\%$ (*C. elegans*) connections were computed that disagree with the opinion of an expert. Even if we assume that errors even out, 2-4% of connections are still completely missing. In the worst case, if all microtubules traverse all sections, this error will add up. For example, in a stack of 20 sections from *X. laevis* the number of lines will be overestimated by 80% in the worst case. Therefore, we recommend to check all unconnected lines before proceeding to analyze the data. This manual correcting takes some time (500/h), but this is still much less work than connecting each line in a dataset. Another benefit of this correction is that checking the line ends can reveal tracing errors (about 4% missing lines, see Weber et al. [100]). Often, when an end was not connected, it turned out to be missing its continuation in the next section. Because the lines are scaled and warped, their lengths might be altered by the registration as well. We have not analyzed how the computed deformations might bias a measurement of microtubule lengths in the resulting network. This is hard to measure, because from inspecting the tomograms we can only infer transformations that register the lines but we cannot assess if this transformation corrects the actual damage on the data.

Note that ground truth for measuring the accuracy was created by letting an expert correct the results of the proposed algorithm pipeline. Our measurements might therefore be biased in favor of the proposed PGM matching approach. Furthermore, in many cases it is hard even for an expert to decide which connections are biologically correct. An example is given in Figure 4.5(c) and 4.5(f).

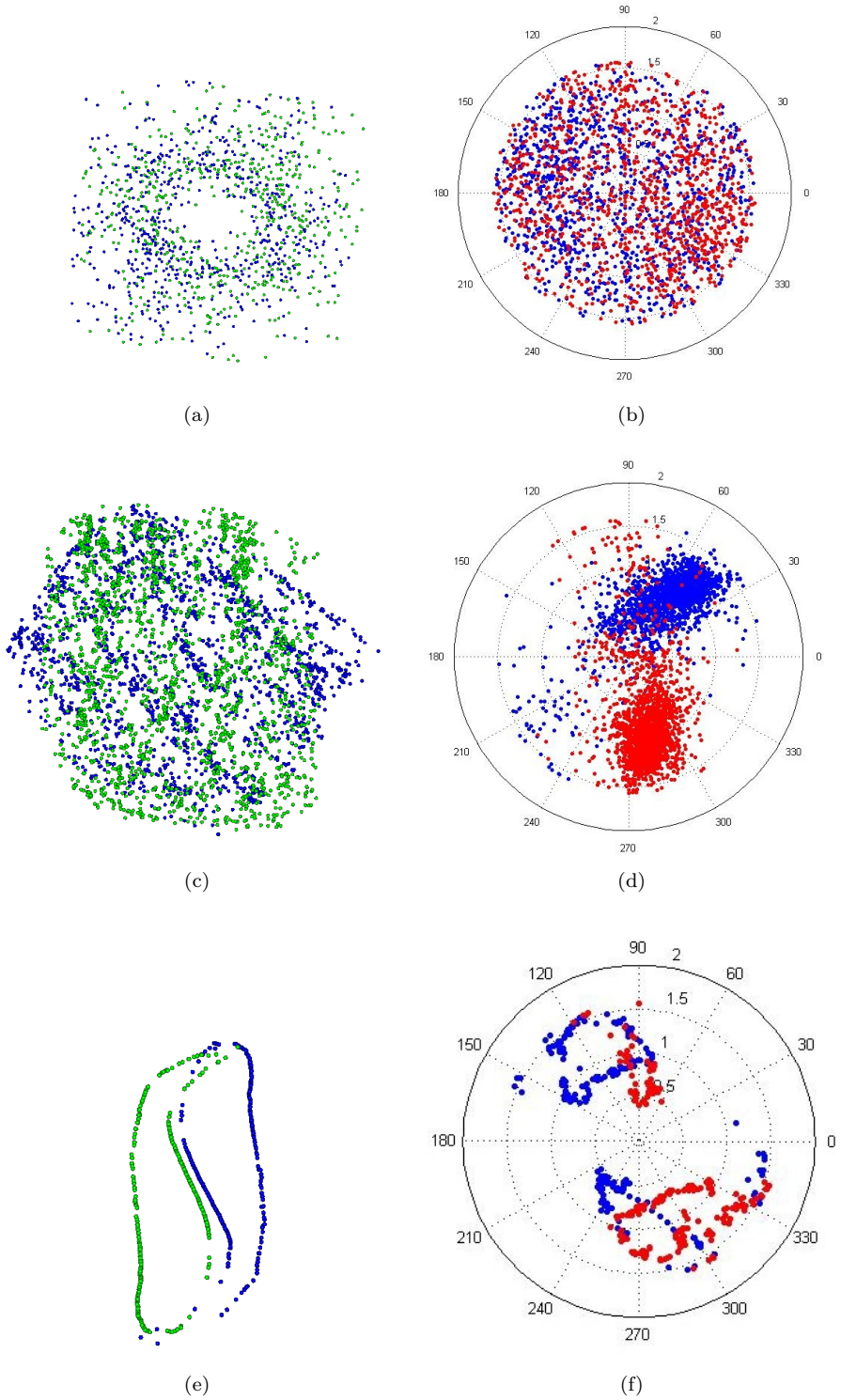
In the end, the major limitation of our pipeline is the initial alignment using the DCG. Though this method worked on all *C. elegans* and *X. laevis* examples, we believe the method will not scale with the number of line ends. Although this is not relevant for microtubules in electron tomography, this may fail for datasets having more than about 4000 lines in each section. The feature descriptor approach described by Preibisch et al. [148] could be a better solution for larger datasets. Another solution might be to replace the initial alignment by a rotation detection using orientation of lines for the data we use. A few different rotations could be used as starting points and the correct result could be automatically chosen by inspecting κ and σ^2.

The proposed pipeline might also help stitching other filamentous structures such as neurons. However, the major assumption for the algorithms is that the lines are rather straight. This is a valid assumption if microtubules are observed in wild type samples, but might not be the case for mutants with altered structural properties or other filamentous structures. For strongly bent curves, a measure taking into account higher order properties such as curvature of lines should be employed instead of only using tangent vectors on lines. Even if the lines are straight, complications might arise from other data modalities that the algorithms do not anticipate. For example, we believe that computing a matching for the *T. brucei* data would require a different approach than the one proposed in this pipeline (PGM). This method requires an expert to decide on correct

pairing of line ends. This is challenging for the *T. brucei* dataset because of the periodic nature of the data (the sub-pellicular microtubule array consists of equidistant microtubules that underlays the entire plasma membrane, Figure 4.17(c) and 4.17(d)).

We believe that aligning tomograms using microtubules as features is much more accurate than using few manually selected landmarks for alignment. Running the proposed automatic stitching might be useful even for small datasets that could be stitched manually. Furthermore, the result of the automatic registration is reproducible, whereas a manual alignment is not. Therefore, if tomograms contain microtubules that are evenly spread in the tomogram, we believe that it would make sense to run a microtubule segmentation and the proposed stitching algorithms afterwards even if the microtubules are not the focus in the experiment.

In summary, we have developed a tool to robustly stitch segmented microtubule centerlines. We believe that this software will speed up, improve and enable scientific output in the field of microtubule cytoskeleton architecture. So far, the analysis of microtubule networks in electron tomograms is mostly based on observations in single sections or specimens containing only a few hundred microtubules. A quantitative analysis of length or number of microtubules was not possible for samples containing thousands of microtubules such as the spindles of *C. elegans* and *X. laevis*. This method finally allows an analysis of the three-dimensional microtubule networks in these large structures.



(a) (b)

(c) (d)

(e) (f)

Figure 4.14. Left row show the coordinates of the endpoints. The right column shows the orientations of lines in a section. Unit vectors are plotted as polar coordinates. Different color indicates to which section the lines belong. (a) and (b) show coordinates and orientations of two sections of a *C. elegans* centrosome, (c) and (d) show the same for two *X. laevis* sections and Figure (e) and (f) coordinates and orientations for two *T. brucei* sections. Datasets were rotated to visualize the patterns formed by coordinates and orientations in the serial sections.

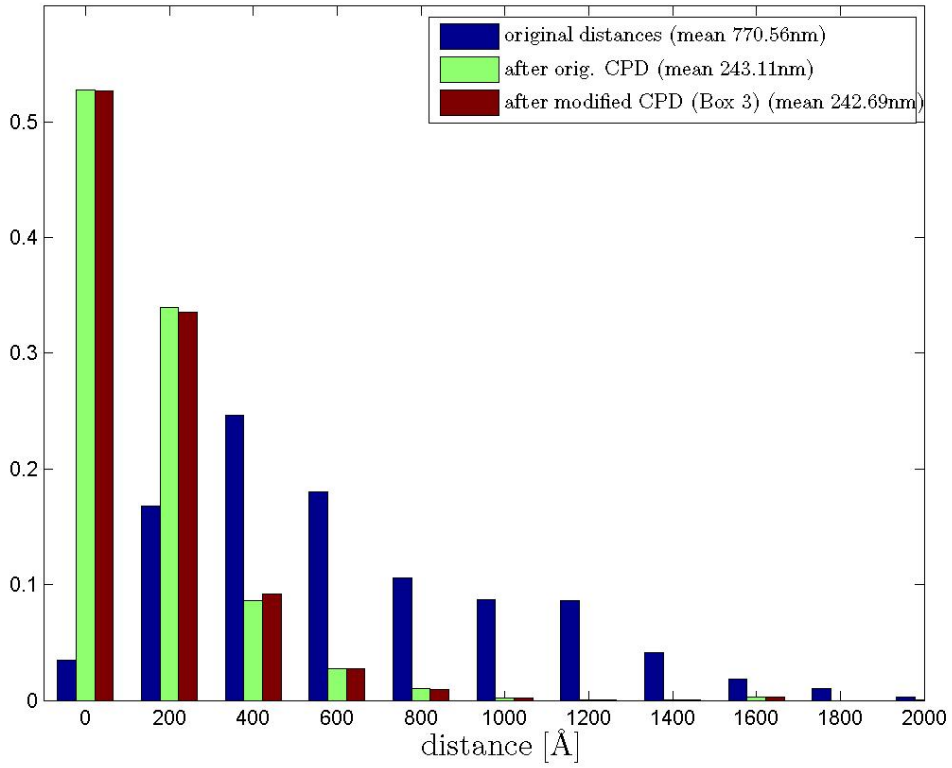
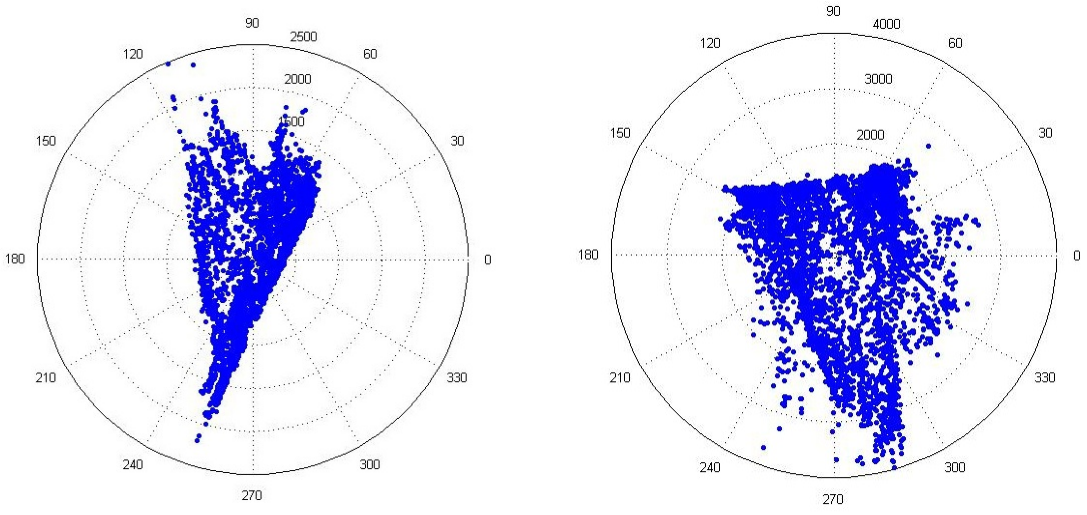
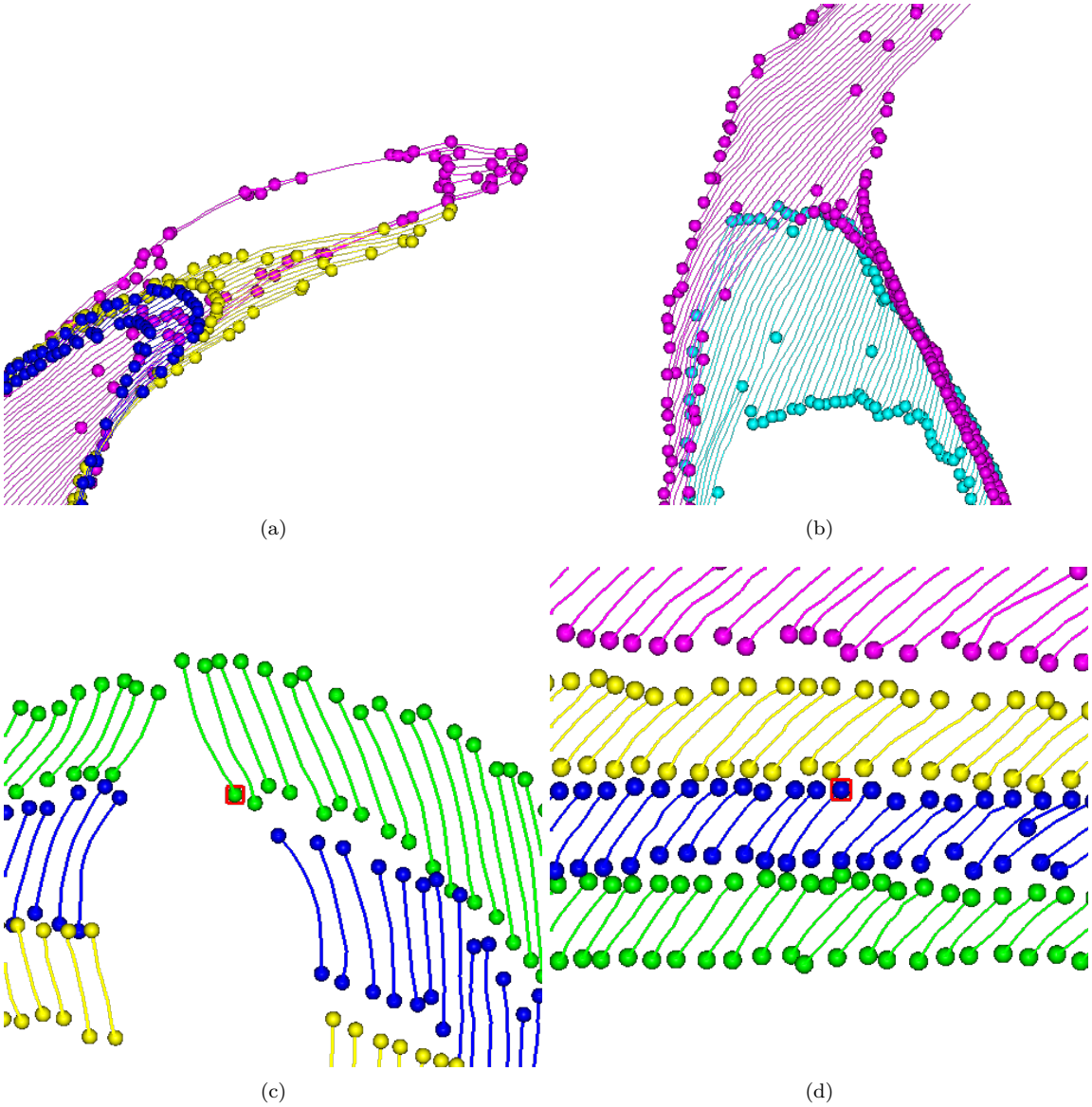


Figure 4.15. Histogram of the distances between matching line ends in the *X. laevis* ground truth before and after aligning with CPD as described in Section 4.4.2.



(a) (b)

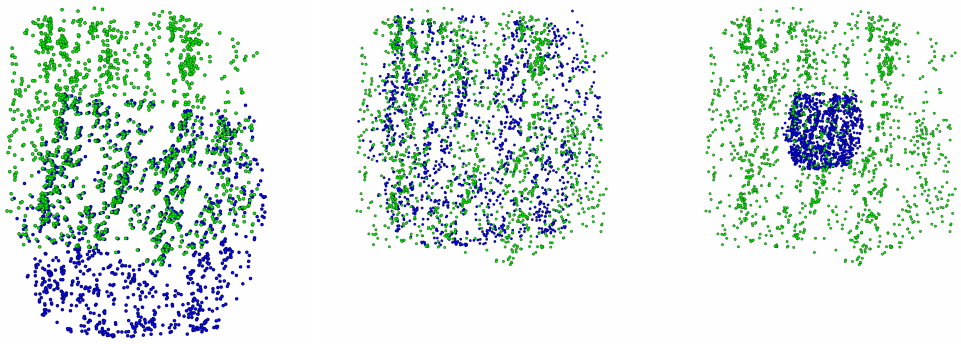
Figure 4.16. Computed displacements for two *X. laevis* section alignments plotted as polar coordinates where azimuth is the orientation of the displacement and elevation the length of the displacement vector.



(a) (b)

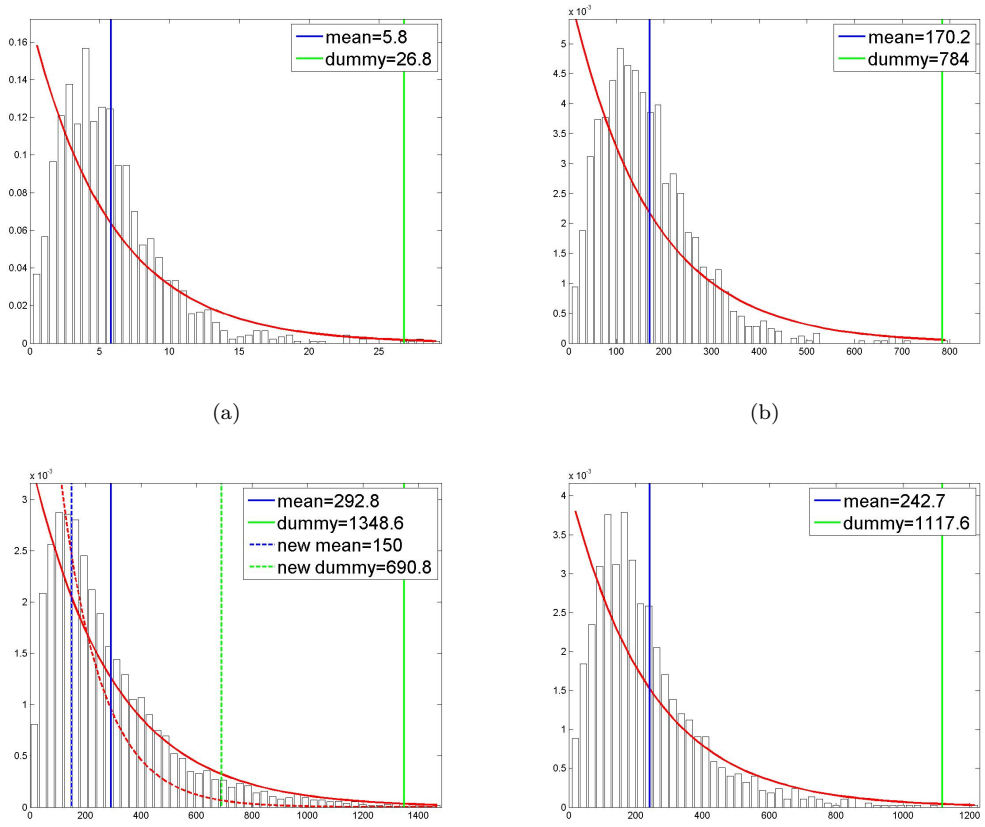
(c) (d)

Figure 4.17. Figures (a) and (b) show the resulting *T. brucei* dataset after elastic registration (Box 4.3). Here, the registration failed to align the sections correctly because the position of ends do not trace out the same shape in the adjacent sections due to strong length fluctuations of the lines. Red nodes in Figures (c) and (d) show examples of maximum disagreement nodes that an expert would have to assign manually to a counterpart in the next Section when using the proposed algorithm for point matching. The right correspondences cannot be identified here without inspecting the full dataset.



(a) (b) (c)

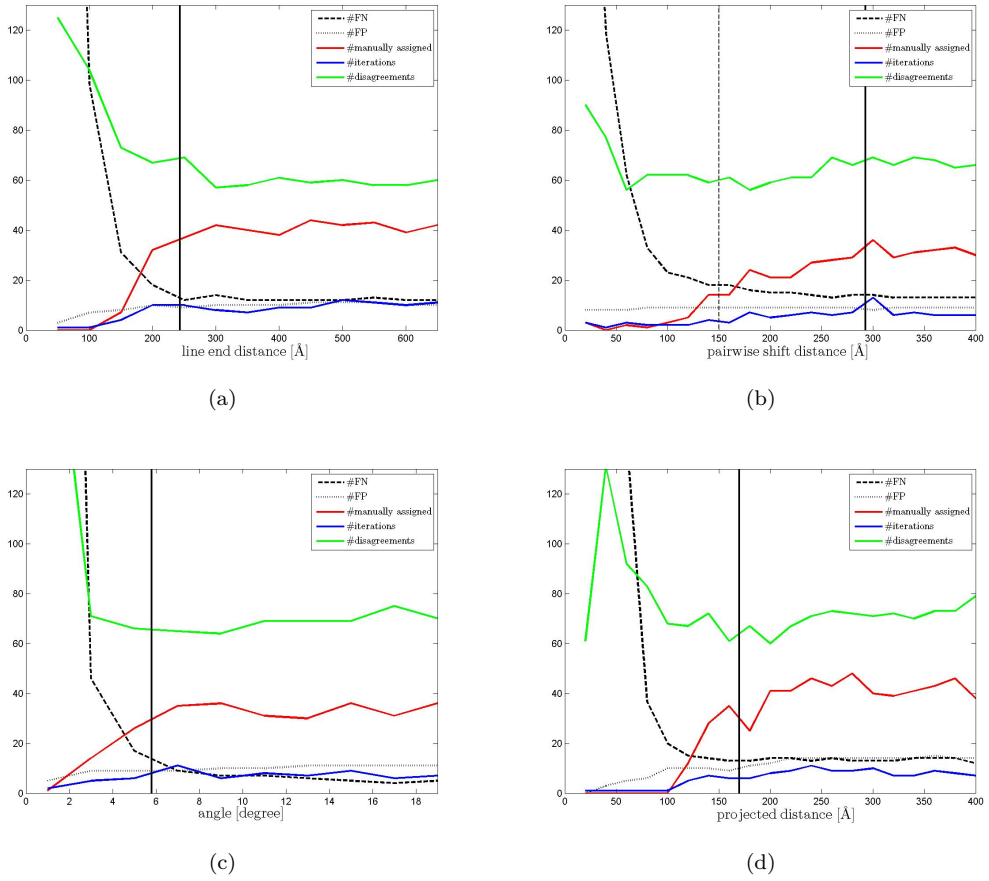
Figure 4.18. Figure (a) shows an example of two line sets with little overlap. (b) and (c) shows the result computed by the rigid alignment as described in [17] and the registration algorithm in Box 4.2 (Section 4.4.2). None of the proposed methods for registration yielded a reasonable result.



(a) (b)

(c) (d)

Figure 4.19. Plots show a normalized histogram of (a) mutual angle d_α, (b) projected distance d_p, (c) shift difference d_s and (d) coordinate distance d_c as obtained by analyzing connections that were checked by an expert. Red vertical line: Estimated mean λ. Red curve: exponential distribution $\lambda \exp(-\lambda x)$ with $\lambda = \frac{1}{\text{mean}}$. Green vertical line: Dummy parameter d^0 computed by $d^0 = -\log(0.01)/\lambda$. In (c), dashed lines indicate the values for the chosen $\lambda = 150$.



(a) (b)

(c) (d)

Figure 4.20. Plots FN, disagreements, number of iterations and number of manual assignments needed for convergence of belief propagation when varying the (a) coordinate distance parameter λ_c, (b) shift difference parameter λ_s, (c) angle difference parameter λ_α and (d) projected distance parameter λ_p about their maximum-likelihood estimate (indicated by solid vertical bar). The dashed vertical bar in Figure (b) indicates the value we chose for λ_s after seeing this plot.

Chapter 5

Microtubule arrangement in the centrosome

In the last chapter, we described a method to stitch stacks of microtubule sections. In this section, we exemplarily discuss potential ways to analyze the resulting lines. As an example, we chose microtubules at a centrosome in a *C. elegans* mitotic spindle (see Section 2.1).

We wrote before that a full reconstruction of all microtubule centerlines from electron tomograms could help answering questions about the organization of the microtubules. However, it is not immediately apparent how to derive shape properties from the line network. For example, for comparing several samples, one requirement is that we derive numerical values from the line representation which can be used to compare one sample to another. The main question here is: What do we actually want to measure?

Several authors have presented methods to analyze different properties, for example, the structure of intepolar microtubules if their polarity is known (Mastronarde et al. [174]), the count of microtubules of a particular polarity (see O'Toole et al. [7]) or the mutual distance of microtubules (Lacomble et al. [133], Höög et al. [5], [134], McDonald et al. [135], Ding et al. [8]). However, the analysis of the full centrosome using electron tomography remains an open issue simply because full reconstructions of the microtubule network were not available until now. Using light microscopy, centrosomes can be visualized by tagging γ-tubulin, the nucleation site of microtubules (Section 2.1), with a fluorescence marker (see, for example, Greenan et al. [13]). This can provide a rough estimate of the centrosome's shape but results are still to be interpreted with care because it is not clear to which extend γ-tubulin surrounds the pericentriolar material (PCM). Furthermore, the organization and properties of the microtubules emerging from the centrosome can not be analyzed with light microscopy because of the limited resolution.

In this chapter, we analyze the shape of the centrosome using a line representation of the microtubules and show how

- the size of the centrosome,
- the number of microtubules emerging from the centrosome and
- the density of microtubules in the centrosome

can be computed from it.

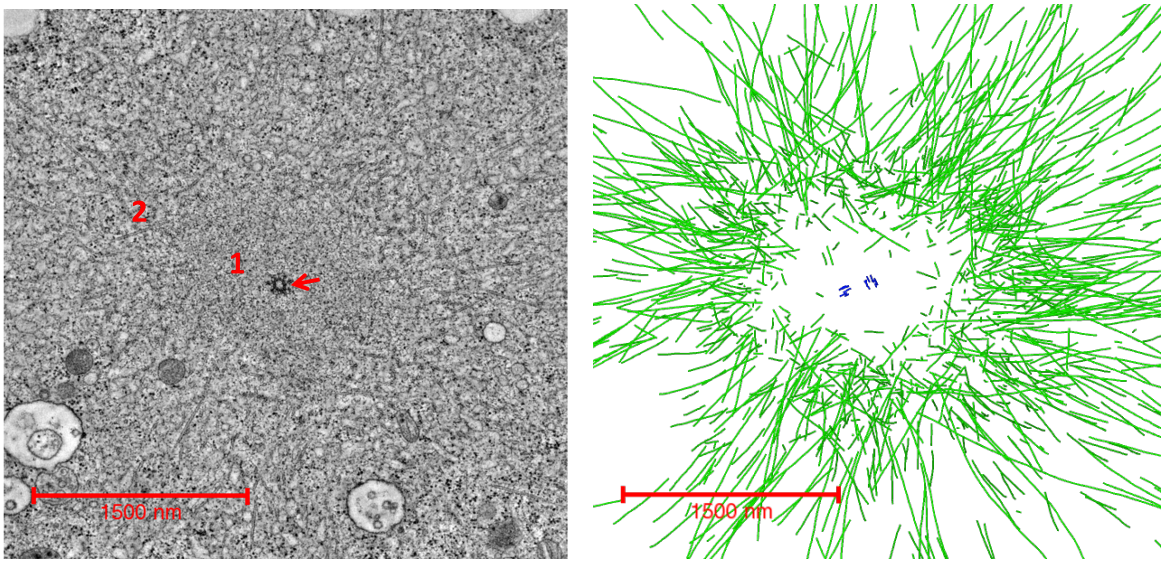
First, we estimate the centrosome's shape using the ends of segmented microtubules that surround the PCM (Figure 5.1) as indicator for the centrosome boundary. To measure the centrosome size, we fit an ellipsoid to these ends (Section 5.2.1).

As for the counting of microtubules, we note that their absolute number in the stack does not correspond to the actual number of microtubules surrounding the centrosome as illustrated in Figure 5.2(a). Individual sections of the sample are shifted and rotated and therefore the stitched volume does not show a cubic subvolume of the sample but instead a volume enclosed in a more complex polyhedral surface. Microtubules miss their continuation in the next section if this next section does not overlap (Figure 5.2(b)). To assure that we do not count these microtubules twice,

we use the ellipsoid that we estimated for the centrosome to measure the number of microtubules and their density as a function of distance from the centrioles (Section 5.2.2). For the latter, we define density as absolute lengths of microtubules contained in a subvolume. We use thin ellipsoidal shells centered at the centrioles as subvolumes as illustrated in Figure 5.2(c).

We exemplarily derive results from a 3D reconstruction of microtubules that was obtained by applying the tools described in Chapter 3 and 4 to a stack of serial electron tomograms. We chose a sample showing the centrosome of a single-cell *C. elegans* embryo at the first mitotic cell division (P0 cell). In Section 5.3 we present results for our test dataset. We conclude this chapter with a discussion of strengths and weaknesses of the presented methods (Section 5.4).

The biological results given in this section are preliminary results. The stitched lines have not been proof read by an expert and the given numbers on centrosome size and number of microtubules must not be used to draw conclusions on biological phenomenon.



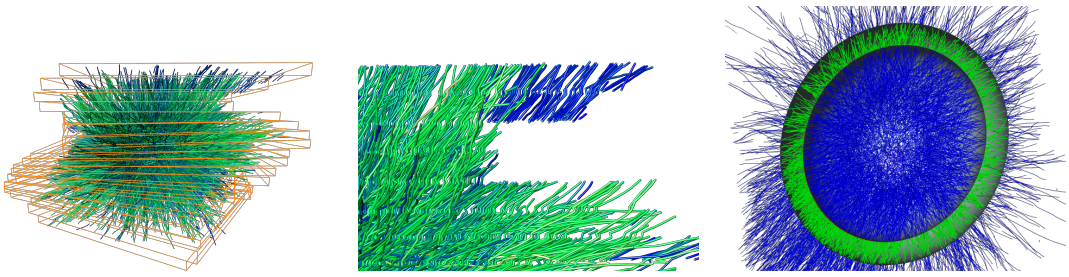
(a) (b)

Figure 5.1. (a) A slice through a tomogram of a centrosome in a *C. elegans* mitotic spindle. The arrow marks the position of the centrioles. The centrosome's (1) shape can be guessed because it has a slightly darker texture than the rest of the tomogram (2). However, it is not delimited by a clear boundary. (b) Microtubules in the same tomogram. The interior of the centrosome is mostly devoid of microtubules (green). Centrioles are depicted in blue.

5.1 Data

We show results for one centrosome of a *C. elegans* mitotic spindle in a P0 cell. The sample was prepared as described in Section 4.5.1. The effective field of view of the camera was about 4.5 μm. 20 sections with a thickness of about 200–300 nm were acquired.

The segmentation of microtubules in each tomogram was performed as described in Section 3 by our collaborators at MPI-CBG, Dresden. The lines were manually proof read by an expert. The sections were stitched as described in Section 4. The final result after point matching was not corrected by an expert. We also did not correct the kinks that often occur in the stitched lines at the junction of two tomograms.



(a) (b) (c)

Figure 5.2. (a) Bounding boxes of tomograms transformed as computed by the linear registration algorithm 2 (Box 4.2). (b) Rotation and shift of the sections cause gaps in the full volume. Color indicates the length. Green: long (several μm), blue: short (few hundred nm). (c) To measure the density of microtubules, we let an ellipse (Section 5.2.1) grow from the center of the centrosome. We define density as the sum of lengths of microtubules between two ellipsoids (green) divided by the volume between two ellipsoids.

5.2 Measurements

5.2.1 Estimating the shape of the centrosome

To estimate the size of the centrosome, we propose to fit an ellipsoid to the microtubule ends that are close to the PCM. Because we cannot judge from the lines alone if an end is a minus end pointing towards the PCM, we use a simple heuristic: For each line, we measure the distance of each end to the bounding box of the stack and pick the end that is further away from the border.

To get center and axis lengths of the ellipse, we fit a 3D Gaussian distribution

$$\mathcal{N}(p) = \frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}} \exp -\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)$$

to the coordinates of the selected end points $\{p_1, ..., p_N\}$. We set the center of the ellipsoid to the mean μ of this Gaussian and define the axes of the ellipsoid as the eigenvectors of the covariance matrix Σ. As for the axes' lengths, we first compute the mean of $\mathcal{N}$ at the line end locations $\{p_1, ..., p_N\}$ as

$$\mu_p = \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(p_i).$$

We then compute the point where $\mathcal{N}$ takes the value μ_p on each axis. To do so, we first notice that the axes' orientations are also the eigenvectors of the inverse covariance matrix, that is $u_j^\top \Sigma^{-1} u_j = \lambda_j^{-1}$. Therefore, we can find the lengths of the axes l_j by solving

$$\mu_p = \frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma|^{\frac{1}{2}}} \exp -\frac{1}{2} l_j^2 \text{ for } l_j \text{ and so}$$

$$l_j = \sqrt{-2\log(\sqrt{|\Sigma|}\mu_p)/\lambda_j}$$

for all three axes.

5.2.2 Measurements on microtubules relative to centrosome shape

Measurements To count microtubules and avoid overestimation as discussed in the introduction, we grow and shrink the ellipsoid by adding a fixed distance d to the axes and measure the number of microtubules that

1. cross the ellipsoid,

2. are contained completely inside the ellipsoid and

3. are completely outside the ellipsoid.

We let the ellipsoid grow until it intersects with the polyhedron that encloses the transformed image data.

Furthermore, we would like to estimate some notion of density of microtubules, because we observed that there is an area devoid of microtubules around the centrioles, a region of high density of microtubules and a decrease in density when moving further away from the centrioles (see Figure 5.1(b)). To do so, we also measure the sum of lengths between two ellipses of different axes lengths as illustrated in Figure 5.2(c). We define the microtubule density with respect to centrosome distance by dividing the sum of lengths by the volume difference of the two ellipsoids at a certain distance.

5.3 Results

5.3.1 Microtubule arrangement and centrosome size

To get an impression of the structural arrangements of microtubules, we first visualize different categories of microtubules. Figure 5.3(b) shows all microtubules shorter than $0.2\,\mu m$ (1122 lines). The short lines cluster in the center of the volume around the centrioles. Figure 5.3(c) shows all microtubules longer than $2\,\mu m$ (2013 lines). Figure 5.3(a) shows the full stack (7241 lines) for comparison.

The ends that were used for estimating the ellipsoid are displayed in green in Figure 5.3(d). The blue ends were not used. Note that we did not correct the label for ends that miss their continuation in the next section (upper right corner, see also Figure 5.2(b)). The ellipsoid that was computed is shown in Figure 5.3(e). The axis lengths were calculated as $1.33/1.05/0.847\,\mu m$. This suggests that most microtubule ends are positioned at a distance of approximately $1\,\mu m$ from the centrioles.

Figure 5.3(f) shows the largest ellipsoid that is still completely inside the polyhedron surface that encloses the transformed tomograms (axes $2.33/2.05/1.85\,\mu m$). Figure 5.3(g) shows all microtubules that were contained completely inside this ellipsoid (2994 lines), Figure 5.3(h) all lines that were crossing the ellipsoid's surface (3647 lines) and Figure 5.3(i) all that were completely outside (600 lines). Note that the latter contains some microtubules that result from the tomograms rotation (Figure 5.2(b)) and can be excluded when counting the absolute number of microtubules. However, some can not be explained by this effect (red arrow). These may indicate that microtubule nucleation is not symmetric around the centrioles.

5.3.2 Size, density and number of microtubules in P0

We changed the axes lengths of the originally fitted ellipsoid (Figure 5.3(e)) by adding a fixed distance d. d ranged from -0.6 μm to $1.5\,\mu m$ in $0.1\,\mu m$ steps. Figure 5.4(a) shows the density we estimated for the different step sizes. Here, we plotted the density $[1/\,\mu m^{\,2}]$ over axis difference $d + 0.05\,\mu m$ of the original ellipse. The curve reaches a peak close to the original ellipse (d=-$0.05\,\mu m$).

Figure 5.4(b) shows the number of microtubules outside (blue), inside (red) and crossing (black) the ellipsoid plotted over d. These measurements are only reliable for $d < 1\,\mu m$ because here larger ellipsoids intersect with the volumes boundary. The number of microtubules crossing the ellipsoid is rather stable between d =0.4 to $1\,\mu m$ with approximately 3800 crossing microtubules.

5.4 Discussion

Analysis methods We have presented ways to measure properties of the microtubules at a centrosome that allow a comparison of different centrosomes, for example, in different stages of cell division or for different types of cells.

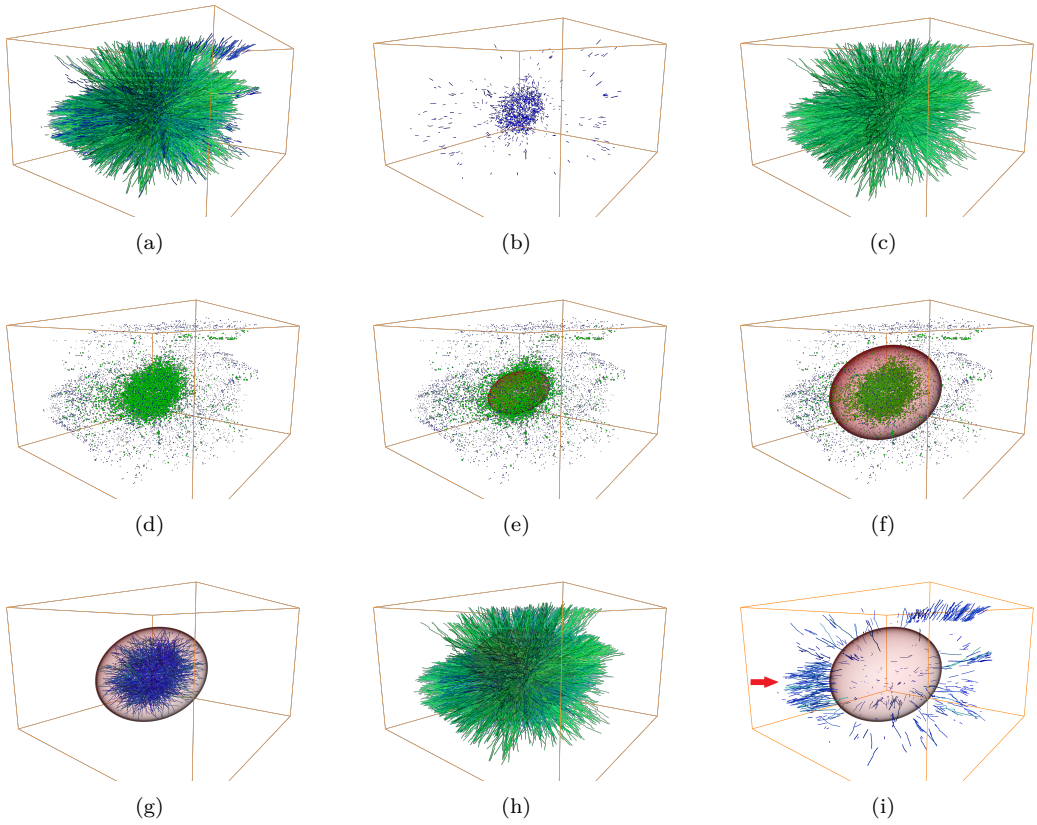


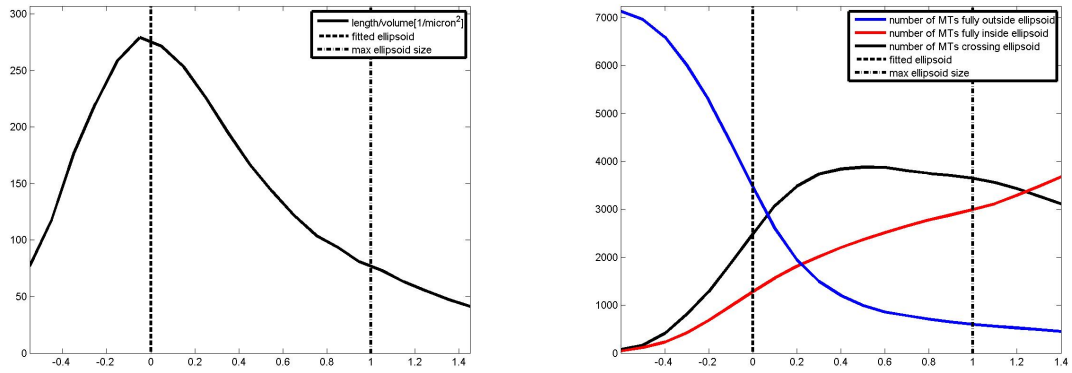
Figure 5.3. (a) The network of microtubules (7241 lines). (b) All microtubules shorter than $0.2\,\mu m$ (1122 lines). (c) All microtubules longer than $2\,\mu m$ (2013 lines). (d) green/blue: Ends chosen/not chosen for fitting the ellipsoid. (e) Fitted ellipsoid (axis $1.33/1.05/0.847\,\mu m$). (f) Largest ellipsoid still contained in volume (axis $2.33/2.05/1.85\,\mu m$). (g) All microtubules inside largest ellipsoid (2994 lines). (h) All microtubules crossing largest ellipsoid (3647 lines). (i) All microtubules outside largest ellipsoid (600 lines). Color bar ranges from $0.2\,\mu m$ (blue) to $5\,\mu m$ (yellow). See Section 5.3.1 for more details. Data courtesy of Jean-Marc Verbavatz and Chris Weiss.

Measuring the density is a reliable measure compared to the absolute number of lines, because breaks have little impact on the sum of lengths as was discussed in Section 3.6.3. Because we computed densities on ellipsoidal shells around the centrosome, small fluctuations on the shell will not be captured. Plotting the density (Figure 5.4(a)) could be used to immediate assess differences between samples.

However, the density measuring proposed in this section can only give a very coarse estimation of the spatial distribution of microtubules in the centrosome. For example, in Figure 5.3(i), we have seen that the assumption of an ellipsoidal shape might not be correct and we cannot assume that the density is constant around the centrosome. To get a realistic measurement of density of microtubules, as would be required for measuring the concentration of polymerized tubulin in the centrosome in certain areas, a more adaptive definition of the subvolumes must be developed.

Further insight can already be gained by the simple ellipsoid fitting we introduced in Section 5.2.1. The axis lengths can be used to determine at which distance from the centrioles microtubules actually nucleate ($1\,\mu m$ in our dataset, see Figure 5.3(e)). The size itself can again be used to compare centrosomes from different species.

Note that we estimated the ellipsoid using the mean of values of a fitted Gaussian to determine the position of the surface instead of directly estimating it by computing the surface that minimizes the distance of points from it. The result for the latter approach might vary slightly from that derived with the proposed method. Furthermore, care must be taken when using size and



(a) (b)

Figure 5.4. (a) Density of microtubules at different distances from the estimated centrosome shape. The density reaches a peak near the size of the originally estimated centrosome ellipsoid (dashed line). (b) Number of microtubules with respect to ellipsoid. See Section 5.3.2 for a detailed description.) Data courtesy of Jean-Marc Verbavatz and Chris Weiss.

density, for example, as input to a biophysical simulation, because shrinking of the sample while microtoming and imaging (Section 2.2.3) was not taken into account yet and therefore the derived numbers might be biased.

We believe that measuring crossings of microtubules with respect to a geometric representation of the centrosome such as an ellipsoid instead of mere counting is less sensitive to breaks as an absolute number of microtubules would be, because a break in a line would only influence the measure locally but not add up. Errors in the tracing or stitching might have a much higher impact on simple counting of microtubules as discussed in Section 4.7. Most importantly, counting the microtubule crossings can give an estimate on how many microtubules actually emerge from the centrosome (about 3800 in the presented dataset, see Figure 5.4(b)).

Finally, a simple visualization of different classes of microtubules (short, long, crossing, etc.) can be used to get an idea of spatial structure in the length distribution of microtubules. For example, we have seen in Figure 5.3(b) that microtubules shorter that 200 nm cluster around the centrosome in our dataset. Figure 5.3(i) showed that microtubule organization is not perfectly symmetric. Here, the red arrow marks an area where the microtubules' ends are spread further from the estimated centrosome boundary than elsewhere. Such structural properties might be interesting when analyzing the functionality of different microtubule types in the centrosome.

We have presented methods to gain insight into and make comparable microtubule organization at the centrosome. The purpose of such an analysis is to gain further insight into the spindle organization. As discussed in the introduction (Section 1.3), the numbers derived with the proposed methods could, for example, be used to validate recent results of microtubule organization in the centrosome such as geometric properties of the spindle or help determining parameters for biophysical simulations. This last step remains to be accomplished.

Chapter 6

Summary and concluding remarks

Electron tomography is currently the only technique that allows visualizing microtubules in cells at a nanometer resolution. Therefore, this imaging technique has the potential to provide detailed insight into the structural arrangement of microtubules in large and dense networks such as the mitotic spindle in *C. elegans* or *X. laevis*. However, due to the difficulty of manually processing the acquired volumes, analysis of microtubules in electron tomograms was restricted to either small structures or individual sections from samples. In this thesis, we have presented methods that allow 1) a segmentation of microtubule centerlines from electron tomograms, and 2) to stitch the lines and finally obtain a 3D reconstruction of all centerlines in a stack of serial electron tomograms.

Tracing of microtubule centerlines The heart of the microtubule tracing is the detection of microtubules by 3D template matching. The templates were designed to resemble the appearance of microtubules in electron tomograms. This included not only an analysis on the template parameters but also a simulation of the missing wedge effect, which is an artifact inherent to electron tomography (Section 3.3.1). We implemented a fast computation of the normalized cross-correlation with the templates using the convolution theorem (Roseman [103]). In addition, we performed the computation on the GPU to finally yield a running time of about 6-12h per tomogram. To trace microtubules, we used the algorithm presented by Rigort et al. [101].

Furthermore, we conducted a thorough evaluation of the method on plastic-embedded sections of *C. elegans* early embryos. First, we analyzed the sensitivity of the parameters of the tracing algorithm. To do so, we compared the result of different parameter settings to a ground truth, which we obtained by combining four manual tracings on the same tomogram to a single tracing (Section 3.5). From this analysis, we obtained a set of parameters for the tracing algorithm. We also used this ground truth to compare the performance of individual users. To validate the computed results, we then tested the performance of the tracing algorithm with the obtained parameter set by comparing the resulting lines to single manually obtained tracings on over 50 electron tomograms of the centrosome of early *C. elegans* embryos.

Our analysis showed that the proposed method is clearly preferable to manual segmentation not only in terms of speed but also in terms of accuracy. However, the achieved 4% false negatives and 8% false positives on average might still prevent a reliable quantitative analysis of, for example, length and number of the lines (Section 3.6). Therefore, we also presented an editor specifically designed for correcting the computed lines. The automatic tracing together with a manual correction significantly reduces the amount of manual labor for tracing microtubule centerlines while at the same time increasing accuracy of the final results.

Stitching stacks of microtubules To obtain a full 3D reconstruction of the microtubules in stacks of serial tomograms of one sample, we developed a pipeline that consists of three steps: 1) An initial coarse alignment which we computed using a distance compatibility graph. 2) A linear and an elastic registration of the line ends computed with a probabilistic variant of iterative closest points. 3) A final matching step that connects the line ends. The matching problem was formulated as a probabilistic graphical model and the matching was finally obtained by computing the maximum a posteriori assignment of the model with belief propagation on a factor graph.

Here, our main contribution is the development of three algorithms for registration of points (step 2) which are associated to a unit vector that mimics the orientation of the microtubules' centerlines. We extended the linear and elastic point registration algorithms presented by Myronenko and Song [17] to take into account a unit vector associated to each point. We introduced a periodic random variable for the unit vectors to the probabilistic framework underlying the methods. We presented an algorithm for a) a linear registration of unit vectors, b) a linear registration of points and vectors and c) an elastic registration of points employing information about the associated unit vectors (Section 4.4.2). These algorithms are variants of an unsupervised learning technique and use the expectation maximization algorithm to obtain final transformation parameters. They are easy to implement and require very few parameter settings.

To compute a matching of the line ends, we formulated the matching problem in terms of a discrete probability distribution as described by Koller and Friedman [162] and Amat et al. [86]. The parameters of this distribution were estimated by analyzing the angle between orientation vectors, distance, projected distance and local shift of points in datasets that were stitched manually. To find an approximate solution for the best matching of ends, we used believe propagation on a factor graph. Because this algorithm does not guarantee convergence, we implemented a user interface which allows enhancement of the original distribution by dedicated user input. We then analyzed how much user input is needed to finally achieve convergence of belief propagation (Section 4.4.3).

Finally, we evaluated the performance of the presented algorithms on biological data from three different species. Our results show that the proposed pipeline is capable of stitching microtubule ends with an error of 5% wrong connections and 5% missing connections in the worst case, while only about 3% of the nodes have to be connected manually (Section 4.6). The error of the automatic stitching might still bias the statistics on length and number. We recommend to post-process unconnected lines manually after automatic stitching was completed.

Results The presented methods allow, for the first time, the analysis of large and dense microtubule networks with electron tomography. We demonstrated the usefulness of the methods by applying the full segmentation and stitching pipeline to a stack of serial tomograms from the centrosome of a *C. elegans* early embryo (Section 5). We estimated the centrosome size, and the number of microtubules emerging from the centrosomes. Such results could serve as input for simulations and as a means to validate simulation models. Furthermore, competing models on organization of microtubules, as discussed in the introduction, can be validated immediately once a full reconstruction of microtubule centerlines is available.

Concluding remarks The key for the success of each of the presented methods is a thorough analysis of the available prior information on the problem and the incorporation of this information into an algorithm.

For the tracing of microtubules, we exploited the detailed knowledge of the shape of microtubules and microscopy specific image modalities. To find the best parameters for the tracing, we learned the optimal parameter set from available manual segmentations. By a detailed adjustments of the methods, we were finally able to obtain a reliable segmentation of the microtubules.

Our approach to the stitching was similar. Here, using probabilistic methods in the stitching step facilitated the integration of expert knowledge. For example, the formulation of the point matching by a probabilistic graphical model allowed us to model our observation that points are shifted coherently in a small neighborhood. Furthermore, we learned reasonable factor entries from biological data. But, most importantly, we found a means to point an expert to connections that were particularly ambiguous and uncertain. We used this specialized knowledge to assure that the computed matching complies with the expert's opinion.

Some of the methods we presented might be applicable to other scenarios. In particular, the registration of line ends (Section 4.4.2) is most likely to be useful for other applications since filamentous structures are often encountered in biology. However, as was already discussed, we believe that the success of each method is based on the careful analysis of available data and subsequent fine tuning of each method to adapt them to the problem at hand. Applying the methods to different data will require a re-configuration of the parameters. Furthermore, rigid assumptions, such as the straightness of microtubules, limit the applicability of the methods.

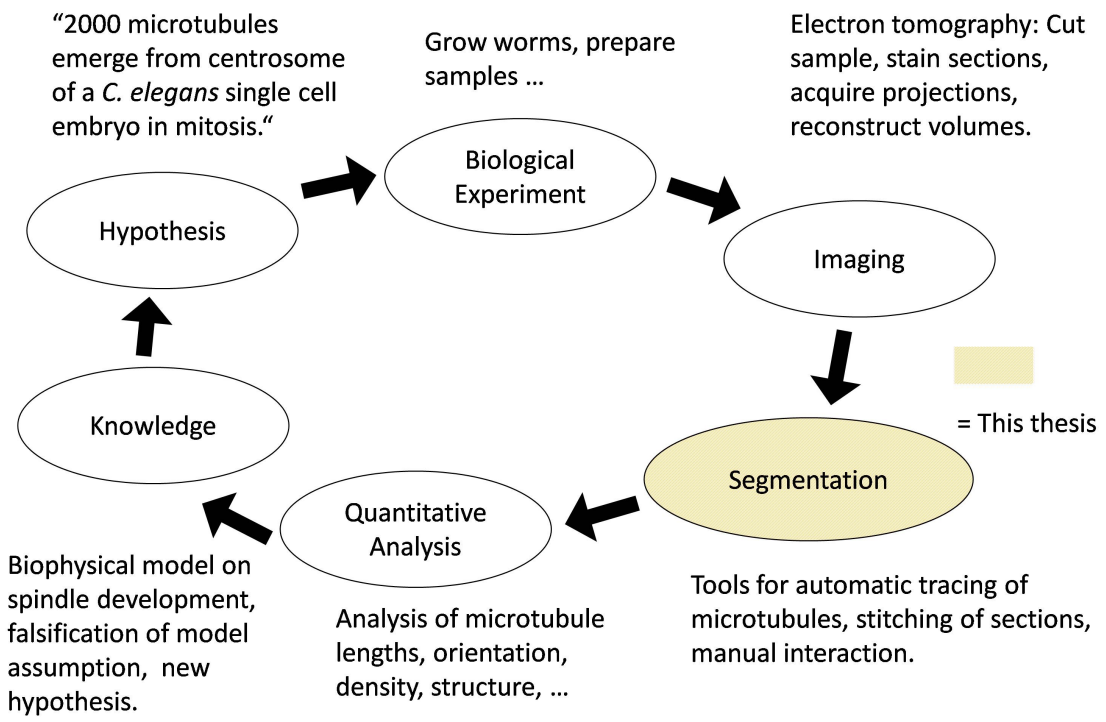


Figure 6.1. The circuit of biological research.

Our work bridges the gap between electron tomography and knowledge about microtubule organization. For the first time, it is possible to analyze microtubule networks containing thousands of microtubules. However, considering the amount of work it takes to obtain a full stack of serial tomograms that cover the sample, it is still not clear whether electron tomography provides the appropriate means to perform this analysis. In the introduction, we discussed that parameters, such as length and number, obtained from the reconstruction could serve as input for biophysical simulations. While a final 3D reconstruction naturally provides detailed insight into these properties, one might wonder whether a different kind of information might yet be more valuable or easier to obtain while serving the same purpose. For example, we have not analyzed how microtubule properties could be derived from incomplete line representations. A thorough statistical analysis of microtubules in single sections of a sample might suffice to estimate parameters such as length and number. Other microscopy techniques might be superior in answering specific questions as well. For example, a fluorescent plus end tracker could also provide information on length of individual microtubules, provided that the resolution allows a distinction of different ends. Finally, light microscopy and scanning electron microscopy are approaching a resolution that might soon be sufficient to resolve individual microtubules. If these new methods succeed, electron tomography will be rendered futile for the analysis of large networks.

The goal of biological research is always refining the model we have about the world. An obtained segmentation of a full microtubule network alone serves no purpose if it does not provide means to this end. The segmentation and stitching methods we presented are just one small step towards understanding. Figure 6.1 depicts a general outline of the role of this work. The project that accompanied the development of this thesis started off with the hypothesis that more than 2000 microtubules emerge from the centrosome of a *C. elegans* mitotic spindle centrosome in P0 (see Müller-Reichert et al. [34]). An experiment was conducted to see whether this assumption holds. The result of the experiment was a stack of electron tomograms. Our contribution in this pipeline is to provide the software for segmenting the microtubule network in order to allow

a quantitative analysis of microtubules properties. However, the last two steps in this pipeline remain: To underpin a theory about an observed phenomenon, the experiment must be repeated several times to gather sufficient evidence. Furthermore, this knowledge again must either be input to a different process, used for validation of a theory, or lead to novel insight which in turn can give rise to new theories.

List of Figures

List of Tables

Bibliography

[1] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, et al. (2007) Molecular Biology of the Cell. Garland Science, Taylor & Francis Group, 5th edition.

[2] Mogilner A, Craig E (2010) Towards a quantitative understanding of mitotic spindle assembly and mechanics. J Cell Sci 123: 3435–3445.

[3] Egerton R (2005) Physical Principles of Electron Microscopy: An Introduction to TEM, SEM, and AEM. Springer.

[4] Weber B, Möller M, Verbavatz JM, Baum D, Hege HC, et al. (2011). Fast Tracing of Microtubule Centerlines in Electron Tomograms. poster. EMBO & NIH Conference Series, Vizualizing Biological Data (VIZBI).

[5] Höög JL, Schwartz C, Noon AT, O'Toole ET, Mastronarde DN, et al. (2007) Organization of Interphase Microtubules in Fission Yeast Analyzed by Electron Tomography. Dev Cell 12: 349–361.

[6] Srayko M, O'Toole ET, Hyman AA, Müller-Reichert T (2006) Katanin Disrupts the Microtubule Lattice and Increases Polymer Number in *C. elegans* Meiosis. Curr Biol 16: 1944–1949.

[7] O'Toole E, Greenan G, Lange KI, Srayko M, Müller-Reichert T (2012) The Role of γ-Tubulin in Centrosomal Microtubule Organization. Plos One 7(1): e29795.

[8] Ding R, McDonald KL, McIntosh JR (1993) Three-dimensional reconstruction and analysis of mitotic spindles from the yeast, Schizosaccharomyces pombe. J Cell Biol 120: 141–151.

[9] Needleman DJ, Farhadifar R (2010) Mitosis: taking the measure of spindle length. Curr Biol 20(8): R359–60.

[10] Yang G, Houghtaling BR, Gaetz J, Liu JZ, Danuser G, et al. (2007) Architectural dynamics of the meiotic spindle revealed by single-fluorophore imaging. Nat Cell Biol 9: 1233–1242.

[11] Karsenti E, Vernos I (2001) The Mitotic Spindle: A Self-Made Machine. Science 294: 543–547.

[12] Wittmann T, Hyman A, Desai A (2001) The spindle: A dynamic assembly of microtubules and motors. Nat Cell Biol 3: E28–E34.

[13] Greenan G, Brangwynne CP, Jaensch S, Gharakhani J, Jülicher F, et al. (2010) Centrosome Size Sets Mitotic Spindle Length in Caenorhabditis elegans Embryos. Curr Biol 20: 353–358.

[14] Paul R, Wollman R, Silkworth WT, Nardi IK, Cimini D, et al. (2009) Computer simulations predict that chromosome movements and rotations accelerate mitotic spindle assembly without compromising accuracy. P Natl Acad Sci USA 106: 15708–15713.

[15] Loughlin R, Heald R, Nédélec F (2010) A computational model predicts Xenopus meiotic spindle organization. J Cell Biol 191: 1239–1249.

[16] Mogilner A, Wollman R, Civelekoglu-Scholey G, Scholey J (2006) Modeling mitosis. Trends Cell Biol 16: 88–96.

[17] Myronenko A, Song X (2010) Point Set Registration: Coherent Point Drift. IEEE T Pattern Anal 32: 2262–2275.

[18] Howard J, Hyman AA (2007) Microtubule polymerases and depolymerases. Curr Opin Cell Biol 19: 31–35.

[19] Akhmanova A, Steinmetz MO (2008) Tracking the ends: a dynamic protein network controls the fate of microtubule tips. Nat Rev Mol Cell Bio 9: 309–22.

[20] Margolin G, Gregoretti IV, Cickovski TM, Li C, Shi W, et al. (2011) The Mechanisms of Microtubule Catastrophe and Rescue: Implications from analysis of a dimer-scale computational model. Mol Biol Cell 23: 642–56.

[21] Kueh HY, Mitchison TJ (2009) Structural Plasticity in Actin and Tubulin Polymer Dynamics. Science 325: 960–963.

[22] Farr H, Gull K (2012) Cytokinesis in trypanosomes. Cytoskeleton (Hoboken) 69: 931–941.

[23] Gull K (2003) Host-parasite interactions and trypanosome morphogenesis: a flagellar pocketful of goodies. Curr Opin Microbiol 6: 365–370.

[24] Nigg EA, Stearns T (2011) The centrosome cycle: Centriole biogenesis, duplication and inherent asymmetries. Nat Cell Biol 13: 1154–1160.

[25] Bao Z, Zhao Z, Boyle TJ, Murray JI, Waterston RH (2008) Control of cell cycle timing during *C. elegans* embryogenesis. Dev Biol 318: 65–72.

[26] Woodruff JB, Drubin DG, Barnes G (2011) Spindle assembly requires complete disassembly of spindle remnants from the previous cell cycle. Mol Biol Cell 23: 258–267.

[27] Burbank KS, Mitchison TJ (2006) Microtubule dynamic instability. Curr Biol 16: R516–R517.

[28] Mitchison T, Kirschner M (1984) Dynamic instability of microtubule growth. Nature 312: 237–242.

[29] Heald R, Nogales E (2002) Microtubule dynamics. J Cell Sci 115: 3–4.

[30] Gouveia SM, Akhmanova A (2010) Chapter One - Cell and Molecular Biology of Microtubule Plus End Tracking Proteins: End Binding Proteins and Their Partners. Academic Press, volume 285 of *Int Rev Cel Mol Bio*. pp. 1–74.

[31] Gell C, Bormuth V, Brouhard GJ, Cohen DN, Diez S, et al. (2010) Chapter 13 - Microtubule Dynamics Reconstituted In Vitro and Imaged by Single-Molecule Fluorescence Microscopy. In: Wilson L, Correia JJ, editors, Microtubules, in vitro, Academic Press, volume 95 of *Method Cell Biol*. pp. 221–245.

[32] Gardner MK, Zanic M, Howard J (2013) Microtubule catastrophe and rescue. Curr Opin Cell Biol 25: 14–22.

[33] Srayko M, Kaya A, Stamford J, Hyman AA (2005) Identification and Characterization of Factors Required for Microtubule Growth and Nucleation in the Early *C. elegans* Embryo. Dev Cell 9: 223–236.

[34] Müller-Reichert T, Greenan G, O'Toole E, Srayko M (2010) The elegans of spindle assembly. Cell Mol Life Sci 67: 2195–2213.

[35] Paweletz N (2001) Walther Flemming: pioneer of mitosis research. Nat Rev Mol Cell Bio 2: 72–75.

[36] Flemming W (1882) Zellsubstanz, Kern und Zelltheilung. Leipzig, F. C. W. Vogel.

[37] Wilson EB (1900) The Cell in Development and Inheritance. Columbia University Biological Series. Macmillan.

[38] Bajer A, Molè-Bajer J (1972) Spindle dynamics and chromosome movements. Academic Press, volume 34 of *Int Rev Cytol*. pp. 1–271.

[39] Inoué S (1953) Polarization optical studies of the mitotic spindle. Chromosoma 5: 487–500.

[40] Inoué S (2008) Microtubule Dynamics in Cell Division: Exploring Living Cells with Polarized Light Microscopy. Annu Rev Cell Dev Bi 24: 1–28.

[41] Mitchison TJ, Salmon ED (2001) Mitosis: a history of division. Nat Cell Biol 3: E17–21.

[42] Finkelstein JM (2008). (1981-1982) Differential Interference Contrast Microscopy: I Can See Clearly Now. Online; accessed 15 June 2013. URL `http://www.nature.com/milestones/milecyto/full/milecyto13.html`.

[43] Allen RD, Allen NS, Travis JL (1981) Video-enhanced contrast, differential interference contrast (AVEC-DIC) microscopy: A new method capable of analyzing microtubule-related motility in the reticulopodial network of allogromia laticollaris. Cell Motil 1: 291–302.

[44] Inoué S (1981) Video image processing greatly enhances contrast, quality, and speed in polarization-based microscopy. J Cell Biol 89: 346–356.

[45] Cassimeris L, Pryer NK, Salmon ED (1988) Real-time observations of microtubule dynamic instability in living cells. J Cell Biol 107: 2223–2231.

[46] Rusk N (2009). First fluorescence microscope, First epifluorescence microscope, The dichroic mirror. Online; accessed 15 June 2013. URL `http://www.nature.com/milestones/milelight/full/milelight04.html`.

[47] Coons AH, Creech HJ, Jones RN (1941) Immunological properties of an antibody containing a fluorescent group. P Soc Exp Biol Med 47: 200–202.

[48] Semenova I, Rodionov V (2007) Fluorescence Microscopy of Microtubules in Cultured Cells. volume 137 of *Methods Mol Med*. pp. 93–102.

[49] Taylor DL, Wang YL (1980) Fluorescently labelled molecules as probes of the structure and function of living cells. Nature 284: 405–410.

[50] Goodson HV, Dzurisin JS, Wadsworth P (2010) Methods for Expressing and Analyzing GFP-Tubulin and GFP-Microtubule-Associated Proteins. Cold Spring Harb Protoc 9: pdb.top85.

[51] Magde D, Elson E, Webb WW (1972) Thermodynamic Fluctuations in a Reacting System—Measurement by Fluorescence Correlation Spectroscopy. Phys Rev Lett 29: 705–708.

[52] Straube A (2011) How to Measure Microtubule Dynamics? In: Straube A, editor, Microtubule Dynamics, Humana Press, volume 777 of *Methods in Molecular Biology*. pp. 1–14.

[53] Boveri T (1888) Zellen-Studien. Jenaische Zeitschrift für Naturwissenschaft .

[54] Matov A, Applegate K, Kumar P, Thoma C, Krek W, et al. (2010) Analysis of microtubule dynamic instability using a plus-end growth marker. Nat Methods 7: 761–768.

[55] Bates M, Huang B, Dempsey GT, Zhuang X (2007) Multicolor Super-Resolution Imaging with Photo-Switchable Fluorescent Probes. Science 317: 1749–1753.

[56] L GMG (2005) Nonlinear structured-illumination microscopy: Wide-field fluorescence imaging with theoretically unlimited resolution. P Natl Acad Sci USA 102: 13081—13086.

[57] Rust MJ, Bates M, Zhuang X (2006) Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). Nat Methods 3: 793–795.

[58] Betzig E, Patterson GH, Sougrat R, Lindwasser OW, Olenych S, et al. (2006) Imaging Intracellular Fluorescent Proteins at Nanometer Resolution. Science 313: 1642–1645.

[59] Hell SW (2007) Far-Field Optical Nanoscopy. Science 316: 1153–1158.

[60] Fernández-Suárez M, Ting AY (2008) Fluorescent probes for super-resolution imaging in living cells. Nat Rev Mol Cell Bio 9: 929–943.

[61] Williams DB, Carter CB (1996) Scattering and Diffraction. In: Transmission Electron Microscopy: Basics, Plenum Press, volume 1, chapter 1. pp. 19–33.

[62] Zhang ZQ, Jones IP, Schriemer HP, Page JH, Weitz DA, et al. (1999) Wave transport in random media: The ballistic to diffusive transition. Phys Rev E Stat Phys Plasmas Fluids Relat Interdiscip Topics 60: 4843–4850.

[63] Frank J (2006) Electron Tomography: Methods for Three-Dimensional Visualization of Structures in the Cell. Springer, 2nd edition.

[64] McIntosh J (2007) Cellular Electron Microscopy. Method Cell Biol. Academic Press.

[65] Knoll M, Ruska E (1932) Das Elektronenmikroskop. Z Phys 78: 318–339.

[66] Harven E, Bernhard W (1956) Etude au microscope electronique de l'ultrastructure du centriole chez les vertébrés. Z Zellforsch Mik Ana 45: 378–398.

[67] Slautterback DB (1963) Cytoplasmic microtubules. Hydra J Cell Biol 18: 367–388.

[68] Sabatini DD, Bensch K, Barrnett RJ (1963) Cytochemistry and Electron Microscopy: The Preservation of Cellular Ultrastructure and Enzymatic Activity by Aldehyde Fixation. J Cell Biol 17: 19–58.

[69] McDonald KL, Auer M (2006) High-pressure freezing, cellular tomography, and structural cell biology. Biotechniques 41: 137–143.

[70] Taylor KA, Glaeser RM (2008) Retrospective on the early development of cryoelectron microscopy of macromolecules and a prospective on opportunities for the future. J Struct Biol 163: 214–223.

[71] Vanhecke D, Studer L, Studer D (2007) Cryoultramicrotomy. In: Kuo J, editor, Electron Microscopy, Humana Press, volume 369 of *Methods Mol Biol.* pp. 175–197.

[72] Marko M, Hsieh C, Schalek R, Frank J, Mannella C (2007) Focused-ion-beam thinning of frozen-hydrated biological specimens for cryo-electron microscopy. Nat methods 4: 215–217.

[73] Buser C, Wather P (2008) Freeze-substitution: the addition of water to polar solvents enhances the retention of structure and acts at temperatures around –60°C. J Microsc 230: 268–277.

[74] Saalfeld S, Fetter R, Cardona A, Tomancak P (2012) Elastic volume reconstruction from series of ultra-thin microscopy sections. Nat Methods 9: 717–720.

[75] Müller HO, Driest E (1935) Elektronenmikroskopische Aufnahmen (Elektronenmikrogramme) von Chitinobjekten. Z Wiss Mikroskopie 52: 53–57.

[76] Ruska E (1993) The Development of the Electron Microscope and of Electron Microscopy. In: Frängsmyr T, Ekspång G, editors, Nobel lectures. 1981/90, World Scientific Publishing Co, Nobel Lectures, Including Presentation Speeches and Laureate.

[77] DeRosier DJ, Klug A (1968) Reconstruction of Three Dimensional Structures from Electron Micrographs. Nature 217: 130–134.

[78] Carazo JM, Herman G, Sorzano C, Marabini R (2006) Algorithms for Three-dimensional Reconstruction From the Imperfect Projection Data Provided by Electron Microscopy. In: Frank J, editor, Electron Tomography, Springer New York. pp. 217–243.

[79] Rademacher M (2006) Weighted Back-projection Methods. In: Frank J, editor, Electron Tomography, Springer New York. pp. 245–274.

[80] Radon J (1986) On the determination of functions from their integral values along certain manifolds. IEEE T Med Imaging 5: 170–176.

[81] Zeitler E (2006) Reconstruction with Orthogonal Functions. In: Frank J, editor, Electron Tomography, Springer New York. pp. 275–306.

[82] Penczek PA (2010) Fundamentals of three-dimensional reconstruction from projections. Method Enzymol 482: 1–33.

[83] Schuberth S (2011) High-Performance Tomographic Reconstruction using OpenCL. Diploma thesis, Department of Mathematics and Computer Science, Freie Universität Berlin. Available upon request.

[84] Penczek P, Frank J (2006) Reconstruction with Orthogonal Functions. In: Frank J, editor, Electron Tomography, Springer New York. pp. 275–306.

[85] Taylor KA, Reedy MC, Córdova L, Reedy MK Three-dimensional reconstruction of rigor insect flight muscle from tilted thin sections. Nature 310: 285–91.

[86] Amat F, Moussavi F, Comolli LR, Elidan G, Downing KH, et al. (2008) Markov random field based automatic image alignment for electron tomography. J Struct Biol 161: 260–275.

[87] Mastronarde DN (1997) Dual-Axis Tomography: An Approach with Alignment Methods That Preserve Resolution. J Struct Biol 120: 343–352.

[88] Castaño-Díez D, Scheffer M, Al-Amoudi A, Frangakis AS (2010) Alignator: A GPU powered software package for robust fiducial-less alignment of cryo tilt-series. J Struct Biol 170: 117–126.

[89] Graham LL, Beveridge TJ (1990) Effect of chemical fixatives on accurate preservation of Escherichia coli and Bacillus subtilis structure in cells prepared by freeze-substitution. J Bacteriol 172: 2150–2159.

[90] Studer, Gnaegi (2000) Minimal compression of ultrathin sections with use of an oscillating diamond knife. J Microsc 197: 94–100.

[91] Moore DH, Morgan C, Rose HM (1956) Some effects of the microtome knife and electron beam on methacrylate-embedded thin sections. J Biophys Biochem Cytol 2: 21–8.

[92] Peachey LD (1958) Thin sections. I. A study of section thickness and physical distortion produced during microtomy. J Biophys Biochem Cytol 4: 233–42.

[93] van Marle J, Dietrich A, Jonges K, Jonges R, de Moor E, et al. (1995) EM-tomography of section collapse, a non-linear phenomenon. Microsc Res Tech 31: 311-316.

[94] Luther PK (2006) Sample shrinkage and radiation damage of plastic sections. In: Frank J, editor, Electron Tomography, Springer New York. pp. 17–48.

[95] Luther PK, Lawrence MC, Crowther RA (1988) A method for monitoring the collapse of plastic sections as a function of electron dose. Ultramicroscopy 24: 7–18.

[96] Kremer JR, Mastronarde DN, McIntosh J (1996) Computer Visualization of Three-Dimensional Image Data Using IMOD. J Struct Biol 116: 71–76.

[97] Hammell CM, Hannon GJ (2012) Inducing RNAi in *C. elegans* by Feeding with dsRNA-expressing E. coli. Cold Spring Harb Protoc 2012.

[98] Marsh BJ, Mastronarde DN, Buttle KF, Howell KE, McIntosh JR (2001) Organellar relationships in the Golgi region of the pancreatic beta cell line, HIT-T15, visualized by high resolution electron tomography. P Natl Acad Sci USA 98: 2399-2406.

[99] Marsh B (2006) Toward a 'Visible Cell'... and beyond. Australian Biochemist 37: 5–10.

[100] Weber B, Greenan G, Prohaska S, Baum D, Hege HC, et al. (2012) Automated tracing of microtubules in electron tomograms of plastic embedded samples of Caenorhabditis elegans embryos. J Struct Biol 178: 129–138.

[101] Rigort A, Günther D, Hegerl R, Baum D, Weber B, et al. (2012) Automated segmentation of electron tomograms for a quantitative description of actin filament networks. J Struct Biol 177: 135–144.

[102] Förster F, Hegerl R (2007) Structure determination in situ by averaging of tomograms. Methods Cell Biol 79: 741–767.

[103] Roseman AM (2003) Particle finding in electron micrographs using a fast local correlation algorithm. Ultramicroscopy 94: 225–236.

[104] Padfield D (2012) Masked Object Registration in the Fourier Domain. IEEE T Image Process 21: 2706–2718.

[105] Zhu Y, Carragher B, Glaeser RM, Fellmann D, Bajaj R, et al. (2004) Automatic particle selection: results of a comparative study. J Struct Biol : 14.

[106] Lebbink MN, Geerts WJC, van der Krift TP, Bouwhuis M, Hertzberger LO, et al. (2007) Template matching as a tool for annotation of tomograms of stained biological structures. J Struct Biol 158: 327–335.

[107] Schoenmakers R, Yakushevska A, Storms M (2010) Template Matching as a Tool to Analyze Electron Tomograms. Microsc Microanal 16: 1854–1855.

[108] Schaap M, Metz C, van Walsum T, van der Giessen AG, Weustink AC, et al. (2009) Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms. Med Image Anal 13: 701–714.

[109] Sandberg K (2007) Methods for image segmentation in cellular tomography. Methods Cell Biol 79: 769–98.

[110] Glaser JR, Glaser EM (1990) Neuron imaging with neurolucida — A PC-based system for image combining microscopy. Comput Med Imag Grap 14: 307–317.

[111] Smith MB, Li H, Shen T, Huang X, Yusuf E, et al. (2010) Segmentation and tracking of cytoskeletal filaments using open active contours. Cytoskeleton (Hoboken) 67: 693–705.

[112] Sandberg K, Brega M (2007) Segmentation of thin structures in electron micrographs using orientation fields. J Struct Biol 157: 403–415.

[113] Jiang M, Ji Q, McEwen B (2006) Automated extraction of fine features of kinetochore microtubules and plus-ends from electron tomography volume. IEEE T Image Process 15: 2035–2048.

[114] Nurgaliev D, Gatanov T, Needleman DJ (2010) Automated identification of microtubules in cellular electron tomography. Methods Cell Biol 97: 475–495.

[115] Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh E (2008) Querying and mining of time series data: experimental comparison of representations and distance measures. Proc VLDB Endow 1: 1542–1552.

[116] Fangerau J, Höckendorf B, Wittbrodt J, Leitte H (2012) Similarity Analysis of Cell Movements in Video Microscopy. In: IEEE Symposium on Biological Data Visualization (BioVis). IEEE, pp. 69–76.

[117] Gelasca ED, Obara B, Fedorov DG, Kvilekval K, Manjunath BS (2009) A biosegmentation benchmark for evaluation of bioimage analysis methods. BMC Bioinformatics 10: 368.

[118] Jomier J, LeDigarcher V, Aylward SR (2005) Comparison of Vessel Segmentations Using STAPLE. In: Lect Notes Comput Sc. volume 3749, pp. 523–530.

[119] Jain V, Bollmann B, Richardson M, Berger D, Helmstaedter M, et al. (2010) Boundary Learning by Optimization with Topological Constraints. In: Proc CVPR IEEE. pp. 2488–2495.

[120] Helmstaedter M, Briggman KL, Denk W (2011) High-accuracy neurite reconstruction for high-throughput neuroanatomy. Nature Neurosci 14: 1081–1088.

[121] Cole F, Golovinskiy A, Limpaecher A, Barros HS, Finkelstein A, et al. (2008) Where do people draw lines? ACM T Graph 27: 88:1–88:11.

[122] Müller-Reichert T, Srayko M, Hyman A, O'Toole ET, Mcdonald K (2007) Correlative Light and Electron Microscopy of Early Caenorhabditis elegans Embryos in Mitosis. Methods Cell Biol 79: 101–119.

[123] Guizetti J, Schermelleh L, Mäntler J, Maar S, Poser I, et al. (2011) Cortical Constriction During Abscission Involves Helices of ESCRT-III–Dependent Filaments. Science 331: 1616–1620.

[124] Mastronarde DN (2005) Automated electron microscope tomography using robust prediction of specimen movements. J Struct Biol 152: 36–51.

[125] NVIDIA (2011) NVIDIA CUDA Programming Guide 2.0.

[126] Fischer B, Buhmann JM (2003) Path-based clustering for grouping of smooth curves and texture segmentation. IEEE T Pattern Anal 25: 513–518.

[127] Stalling D, Westerhoff M, Hege HC (2005) Amira: a Highly Interactive System for Visual Data Analysis. In: Hansen CD, Johnson CR, editors, The Visualization Handbook, Elsevier, chapter 38. pp. 749–767.

[128] Torsney-Weir T, Saad A, Möller T, Hege HC, Weber B, et al. (2011) Tuner: Principled Parameter Finding for Image Segmentation Algorithms Using Visual Response Surface Exploration. IEEE Trans Vis Comput Graph 17: 1892–1901.

[129] Modersitzki J (2004) Numerical Methods for Image Registration. Oxford University Press.

[130] McIntosh JR, O'Toole E, Zhudenkov K, Morphew M, Schwartz C, et al. (2013) Conserved and divergent features of kinetochores and spindle microtubule ends from five species. J Cell Biol 200: 459–474.

[131] O'Toole ET, Winey M, McIntosh J, Mastronarde DN (2002) Electron tomography of yeast cells. In: Christine Guthrie GRF, editor, Guide to Yeast Genetics and Molecular and Cell Biology Part C, Academic Press, volume 351 of *Method Enzymol.* pp. 81–96.

[132] Dercksen VJ, Weber B, Günther D, Oberlaender M, Prohaska S, et al. (2009) Automatic alignment of stacks of filament data. I S Biomed Imaging : 971 – 974.

[133] Lacomble S, Vaughan S, Gadelha C, Morphew MK, Shaw MK, et al. (2009) Three-dimensional cellular architecture of the flagellar pocket and associated cytoskeleton in trypanosomes revealed by electron microscope tomography. J Cell Sci 122: 1081–1090.

[134] Höög JL, Huisman SM, Sebö-Lemke Z, Sandblad L, McIntosh JR, et al. (2011) Electron tomography reveals a flared morphology on growing microtubule ends. J Cell Sci 124: 693–698.

[135] McDonald KL, O'Toole ET, Mastronarde DN, McIntosh JR (1992) Kinetochore microtubules in PTK cells. J Cell Biol 118: 369–383.

[136] Eggert DW, Lorusso A, Fisher RB (1997) Estimating 3-D rigid body transformations: a comparison of four major algorithms. Mach Vision Appl 9: 272–290.

[137] Umeyama S (1991) Least-Squares Estimation of Transformation Parameters Between Two Point Patterns. IEEE T Pattern Anal 13: 376–380.

[138] Gower JC, Dijksterhuis GB (2004) Procrustes problems, volume 30 of *Oxford Statistical Science Series*. Oxford University Press.

[139] Walker MW, Shao L, Volz RA (1991) Estimating 3-D location parameters using dual number quaternions. CVGIP-Imag Understan 54: 358–367.

[140] Wolberg G (1996) Recent advances in image morphing. In: Computer Graphics International, 1996. Proceedings. IEEE Computer Society, pp. 64–71.

[141] Duchon J (1977) Splines minimizing rotation-invariant semi-norms in Sobolev spaces. In: Schempp W, Zeller K, editors, Constructive Theory of Functions of Several Variables, Springer Berlin Heidelberg, volume 571 of *Lecture Notes in Mathematics*. pp. 85–100.

[142] Wahba G (1990) Spline models for observational data. Society for Industrial and Applied Mathematics (SIAM).

[143] Bookstein FL (1989) Principal warps: thin-plate splines and the decomposition of deformations. IEEE T Pattern Anal 11: 567–585.

[144] Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2007) Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press, 3rd edition.

[145] Schaefer S, McPhail T, Warren J (2006) Image deformation using moving least squares. ACM Trans Graph 25: 533–540.

[146] Li J, Allinson NM (2008) A comprehensive review of current local features for computer vision. Neurocomput 71: 1771–1787.

[147] Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24: 381–395.

[148] Preibisch S, Saalfeld S, Schindelin J, Tomancak P (2010) Software for bead-based registration of selective plane illumination microscopy data. Nat Methods 7: 418–419.

[149] Yao J, Ruggeri M, Taddei P, Sequeira V (2010) Robust range image registration using 3D lines. In: IEEE Image Proc. pp. 4321 -4324.

[150] Baum D, Hege HC (2006) A Point-Matching Based Algorithm for 3D Surface Alignment of Drug-Sized Molecules. In: R Berthold M, Glen R, Fischer I, editors, Computational Life Sciences II, Springer Berlin Heidelberg, volume 4216 of *Lect Notes Comput Sc*. pp. 183–193.

[151] Besl PJ, McKay ND (1992) A Method for Registration of 3-D Shapes. IEEE T Pattern Anal 14: 239–256.

[152] Rusinkiewicz S, Levoy M (2001) Efficient variants of the ICP algorithm. In: Third International Conference on 3-D Digital Imaging and Modeling, 2001. Proceedings. pp. 145–152.

[153] Rangarajan A, Chui H, Mjolsness E, Pappu S, Davachi L, et al. (1997) A robust point-matching algorithm for autoradiograph alignment. Med Image Anal 1: 379–398.

[154] Wells WM III (1997) Statistical Approaches to Feature-Based Object Recognition. Int J Comput Vision 21: 63–98.

[155] Jian B, Vemuri BC (2011) Robust Point Set Registration Using Gaussian Mixture Models. IEEE T Pattern Anal 33: 1633–1645.

[156] Kamgar-Parsi B, Kamgar-Parsi B (2004) Algorithms for Matching 3D Line Sets. IEEE T Pattern Anal 26: 582–593.

[157] Caetano TS, Caelli T, Schuurmans D, Barone DAC (2006) Graphical models and point pattern matching. IEEE T Pattern Anal 28: 1646–1663.

[158] Sanghavi S, Malioutov D, Willsky A (2011) Belief Propagation and LP Relaxation for Weighted Matching in General Graphs. IEEE T Inform Theory 57: 2203–2212.

[159] Hogrebe L, Paiva AR, Jurrus E, Christensen C, Bridge M, et al. (2012) Serial section registration of axonal confocal microscopy datasets for long-range neural circuit reconstruction. J Neurosci Meth 207: 200–210.

[160] Myronenko A, Song XB (2009) On the closed-form solution of the rotation matrix arising in computer vision problems. arXiv .

[161] Bishop CM (2007) Pattern Recognition and Machine Learning. Springer, 1st edition.

[162] Koller D, Friedman N (2009) Probabilistic Graphical Models: Principles and Techniques. MIT Press.

[163] Komodakis N, Tziritas G (2007) Approximate Labeling via Graph Cuts Based on Linear Programming. IEEE T Pattern Anal 29: 1436–1453.

[164] Kschischang FR, Frey BJ, Loeliger HA (1998) Factor Graphs and the Sum-Product Algorithm. IEEE T Inform Theory 47: 498–519.

[165] Bron C, Kerbosch J (1973) Algorithm 457: finding all cliques of an undirected graph. Commun ACM 16: 575–577.

[166] Mardia KV, Jupp PE (2000) Directional statistics. Wiley series in probability and statistics. Chichester: Wiley. Previous ed. published as: Statistics of directional data. London : Academic Press, 1972.

[167] Waechter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math Program 106: 25–57.

[168] Mooij JM (2010) libDAI: A Free and Open Source C++ Library for Discrete Approximate Inference in Graphical Models. J Mach Learn Res 11: 2169–2173.

[169] Elidan G, McGraw I, Koller D (2006) Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing. In: Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI). Boston, Massachussetts.

[170] Weber B, Tranfield E, Prohaska S, Baum D, Höög J, et al. (2013) Automated stitching of microtubule stacks. tba tba: tba.

[171] Höög JL, Gluenz E, Vaughan S, Gull K (2010) Chapter 8 - Ultrastructural Investigation Methods for Trypanosoma brucei. In: Müller-Reichert T, editor, Electron Microscopy of Model Systems, Academic Press, volume 96 of *Method Cell Biol.* pp. 175–196.

[172] Hannak E, Heald R (2006) Investigating mitotic spindle assembly and function in vitro using Xenopus laevis egg extracts. Nat Protoc 1: 2305–2314.

[173] Murray AW (1991) Cell cycle extracts. Methods Cell Biol 36: 581–605.

[174] Mastronarde DN, McDonald KL, Ding R, McIntosh JR (1993) Interpolar spindle microtubules in PTK cells. J Cell Biol 123: 1475–1489.

Appendix A

Technical details for line end registration

A.1 Sum to matrix rules

For two sets of points $x_i, ...x_N$ and $y_i, ...y_M$ the matrix $\mathbf{X}$ contains all points in X as rows $\mathbf{X} = (x_1, .., x_N)^\top$ and likewise $\mathbf{Y} = (y_1, .., y_M)^\top$.

$\mathbf{P}$ denotes an $M \times N$ matrix that contains the values of the posterior $P(\mathbf{m}|\mathbf{x})$ for all possible pairs x_n, y_m. Furthermore, we write $N_P = \sum_{m,n=1}^{M,N} P(\mathbf{m}|\mathbf{x})$.

The operator $\mathrm{diag}(v)$ creates a diagonal matrix from a vector $v \in R^D$ by $\mathrm{diag}(v) = \begin{pmatrix} v_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & v_D \end{pmatrix}$.

We denote a vector containing only ones as $\mathbf{1}$.

$$\sum_{n=1}^{N} x_i^\top x_i = \mathrm{tr}(X^\top X) \tag{A.1}$$

$$\sum_{n=1}^{N} x_i^\top R y_i = \mathrm{tr}(X^\top Y R^\top) \tag{A.2}$$

$$\sum_{n=1}^{N} x_i x_i^\top = X^\top X \tag{A.3}$$

$$\sum_{n=1}^{N} x_i R y_i^\top = X^\top Y R^\top \tag{A.4}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n = \mathbf{X}^\top \mathbf{P}^\top \mathbf{1} \tag{A.5}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) y_n = \mathbf{Y}^\top \mathbf{P} \mathbf{1} \tag{A.6}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n^\top x_n = \operatorname{tr}(\mathbf{X}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\mathbf{X}) \tag{A.7}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n x_n^\top = \mathbf{X}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\mathbf{X} \tag{A.8}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) y_m^\top y_m = \operatorname{tr}(\mathbf{Y}^\top \operatorname{diag}(\mathbf{P1})\mathbf{Y}) \tag{A.9}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) y_m y_m^\top = \mathbf{Y}^\top \operatorname{diag}(\mathbf{P1})\mathbf{Y} \tag{A.10}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n^\top R y_m = \operatorname{tr}(\mathbf{X}^\top \mathbf{P}^\top \mathbf{Y} R^\top) \tag{A.11}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n y_m^\top R^\top = \mathbf{X}^\top \mathbf{P}^\top \mathbf{Y} R^\top \tag{A.12}$$

A.2 Q derivatives

Splitting the negative log-likelihood function from Section 4.4.2.2 (Equation 4.40) again into terms dependent and independent of $\tilde{R}$ (by Equation 4.21), Q can be written as :

$$\begin{aligned} Q(s, \tilde{R}, \sigma^2, \kappa) = \frac{1}{2\sigma^2}(&\operatorname{tr}(\hat{\mathbf{X}}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\hat{\mathbf{X}}) - 2s \cdot \operatorname{tr}((\hat{\mathbf{X}}^\top \mathbf{P}^\top \hat{\mathbf{Y}})^\top \tilde{R}) + s^2 \operatorname{tr}(\hat{\mathbf{Y}}^\top \operatorname{diag}(\mathbf{P1})\hat{\mathbf{Y}})) - \\ &\kappa(\operatorname{tr}((\vec{\mathbf{X}}_{2d}^{\top} \mathbf{P}^\top \vec{\mathbf{Y}}_{2d})^\top \tilde{R}) + \vec{\mathbf{X}}_z^{\top} \mathbf{P}^\top \vec{\mathbf{Y}}_z) \\ &N_P \log \sigma^2 - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) \end{aligned} \tag{A.13}$$

To find the optimal parameter set $\{s, \tilde{R}, \sigma^2, \kappa\}$, we must find the partial derivatives of Q. To make notation easier, we define $A = \hat{\mathbf{X}}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\hat{\mathbf{X}}$, $B = \hat{\mathbf{X}}^\top \mathbf{P}^\top \hat{\mathbf{Y}}$, $C = \hat{\mathbf{Y}}^\top \operatorname{diag}(\mathbf{P1})\hat{\mathbf{Y}}$, $D = \vec{\mathbf{X}}_{2d}^{\top} \mathbf{P}^\top \vec{\mathbf{Y}}_{2d}$ and $E = \vec{\mathbf{X}}_z^{\top} \mathbf{P}^\top \vec{\mathbf{Y}}_z$, and write again Q

$$\begin{aligned} Q(s, \tilde{R}, \sigma^2, \kappa) = \frac{1}{2\sigma^2}(&\operatorname{tr}(A) - 2s \cdot \operatorname{tr}(B^\top \tilde{R}) + s^2 \operatorname{tr}(C)) - \\ &\kappa(\operatorname{tr}(D^\top \tilde{R}) + \operatorname{tr}(E)) + \\ &N_P \log \sigma^2 - N_P \log(\frac{\kappa}{2\pi(e^\kappa - e^{-\kappa})}) + \text{const.} \end{aligned} \tag{A.14}$$

To compute the partial derivative with respect to $\tilde{R}$, we note that $\tilde{R}$ is parametrized via one rotation angle ρ as $\tilde{R} = \begin{pmatrix} \cos\rho & -\sin\rho \\ \sin\rho & \cos\rho \end{pmatrix}$ in the 2d case. Writing the trace as a sum and taking partial derivatives wrt. this angle we find that

$$\frac{\partial \operatorname{tr}(K^\top \tilde{R})}{\partial \rho} = \operatorname{tr}(K^\top \frac{\partial \tilde{R}}{\partial \rho}) \tag{A.15}$$

and

$$\frac{\partial \tilde{R}}{\partial \rho} = \tilde{R}' = \begin{pmatrix} -\sin\rho & -\cos\rho \\ \cos\rho & -\sin\rho \end{pmatrix} \tag{A.16}$$

The second derivative can be found equivalently.

$$\frac{\partial^2 \tilde{R}}{\partial^2 \rho^2} = \tilde{R}'' = \begin{pmatrix} -\cos\rho & \sin\rho \\ -\sin\rho & -\cos\rho \end{pmatrix} \tag{A.17}$$

We give the partial derivatives of eq. A.14 wrt. s, ρ, σ^2 and κ in Table A.1:

$A = \hat{\mathbf{X}}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\hat{\mathbf{X}}$	$B = \hat{\mathbf{X}}^\top \mathbf{P}^\top \hat{\mathbf{Y}}$	$C = \hat{\mathbf{Y}}^\top \operatorname{diag}(\mathbf{P1})\hat{\mathbf{Y}}$	$D = \vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d}$	$E = \vec{\mathbf{X}}_z^\top \mathbf{P}^\top \vec{\mathbf{Y}}_z$

$\partial/\partial s$	$\partial/\partial \rho$	$\partial/\partial \kappa$	$\partial/\partial \sigma^2$	
$\frac{1}{\sigma^2}(-\operatorname{tr}(B^\top \tilde{R}) + s \operatorname{tr}(C))$	$-\frac{s}{\sigma^2}\operatorname{tr}(B^\top \tilde{R}') - \kappa \operatorname{tr}(D^\top \tilde{R}')$	$-\operatorname{tr}(D^\top \tilde{R}) - \operatorname{tr}(E) - N_P(\frac{1}{\kappa} - \coth(\kappa))$	$-\frac{1}{2\sigma^4}(\operatorname{tr}(A) - 2s \cdot \operatorname{tr}(B^\top \tilde{R}) + s^2 \operatorname{tr}(C)) + \frac{N_P}{\sigma^2}$	1
$\frac{1}{\sigma^2}(\operatorname{tr}(C))$	$-\frac{1}{\sigma^2}(\operatorname{tr}(B^\top \tilde{R}'))$	0	$-\frac{1}{\sigma^4}(-\operatorname{tr}(B^\top \tilde{R}) + s \operatorname{tr}(C))$	$\partial/\partial s$
	$-\frac{s}{\sigma^2}\operatorname{tr}(B^\top \tilde{R}'') - \kappa \operatorname{tr}(D^\top \tilde{R}'')$	$-\operatorname{tr}(D^\top \tilde{R}')$	$\frac{s}{\sigma^4}(\operatorname{tr}(B^\top \tilde{R}'))$	$\partial/\partial \rho$
		$-N_P(-\frac{1}{\kappa^2} + \frac{1}{\sinh(\kappa)^2})$	0	$\partial/\partial \kappa$
			$\frac{1}{\sigma^6}(\operatorname{tr}(A) - 2s \cdot \operatorname{tr}(B^\top \tilde{R}) + s^2 \operatorname{tr}(C)) - \frac{N_P}{\sigma^4}$	$\partial/\partial \sigma^2$

Table A.1. Derivatives of Q for minimization in registration algorithm (Box 4.2).