Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

BRITTA WEBER, ERIN M. TRANFIELD, JOHANNA L. HÖÖG, DANIEL BAUM, CLAUDE ANTONY, TONY HYMAN, JEAN-MARC VERBAVATZ, STEFFEN PROHASKA

Automated stitching of microtubule centerlines across serial electron tomograms[1]

[1] the manuscript will appear in possibly slightly revised form in PLOS ONE

ZIB Report 14-41 (October 2014)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Automated stitching of microtubule centerlines across serial electron tomograms

Britta Weber[1,3], Erin M. Tranfield[2], Johanna L. Höög[3,4], Daniel Baum[1], Claude Antony[2], Tony Hyman[3], Jean-Marc Verbavatz[3,*], Steffen Prohaska[1,#]

1 Zuse Institute Berlin, Germany

2 European Molecular Biology Laboratory, Heidelberg, Germany

3 Max Planck Institute for Molecular Biology and Genetics, Dresden, Germany

4 Sir William Dunn School of Pathology, University of Oxford, UK

∗ **E-mail: verbavat@mpi-cbg.de** # **E-mail: prohaska@zib.de**

Abstract

Tracing microtubule centerlines in serial section electron tomography requires microtubules to be stitched across sections, that is lines from different sections need to be aligned, endpoints need to be matched at section boundaries to establish a correspondence between neighboring sections, and corresponding lines need to be connected across multiple sections. We present computational methods for these tasks: 1) An initial alignment is computed using a distance compatibility graph. 2) A fine alignment is then computed with a probabilistic variant of the iterative closest points algorithm, which we extended to handle the orientation of lines by introducing a periodic random variable to the probabilistic formulation. 3) Endpoint correspondence is established by formulating a matching problem in terms of a Markov random field and computing the best matching with belief propagation. Belief propagation is not generally guaranteed to converge to a minimum. We show how convergence can be achieved, nonetheless, with minimal manual input. In addition to stitching microtubule centerlines, the correspondence is also applied to transform and merge the electron tomograms. We applied the proposed methods to samples from the mitotic spindle in *C. elegans*, the meiotic spindle in *X. laevis*, and sub-pellicular microtubule arrays in *T. brucei*. The methods were able to stitch microtubules across section boundaries in good agreement with experts' opinions for the spindle samples. Results, however, were not satisfactory for the microtubule arrays. For certain experiments, such as an analysis of the spindle, the proposed methods can replace manual expert tracing and thus enable the analysis of microtubules over long distances with reasonable manual effort.

Author Summary

Microtubules are part of the cell cytoskeleton. The cytoskeleton is a three-dimensional structure that provides a scaffold for the cell structure and organizes several subcellular organelles. Microtubules also play a critical role in several cell processes, such as cell division and intracellular transport. Due to their proximity and small diameter (25 nm), individual microtubules in dense microtubule assemblies can only be resolved by electron microscopy. Three-dimensional analysis of cell structures by electron microscopy, however, is challenging. In most cases, it involves manual segmentation of a limited number of thin or semi-thin serial sections. In a previous paper, we reported a method for automated segmentation of microtubules in electron tomograms [1]. Here, we present computational methods for stitching microtubule centerlines across serial sections and evaluate the methods on several samples. Results indicate that the methods can be used for tracing microtubule centerlines in three dimensions over long distances, thus enabling analyzing complete biological structures such as full centrosomes or spindles during cell division.

Introduction

The problem of stitching lines arises when filamentous or tubular structures, such as microtubules, need to be traced across multiple consecutive semi-thin (a few hundred nanometer thick) sections, each of which contains only a portion of the filaments. Three-dimensional images of such sections can be obtained, for example, by electron tomography. Due to sample processing, however, the sections are often individually rotated, shifted, scaled, or deformed in a non-linear way. In this paper, we focus on the stitching of microtubules segmented from serial section electron tomograms. Within each section, microtubules are represented as polygonal centerlines. For microtubules that traverse the entire thickness of the section, the lines have two endpoints at or close to the two section boundaries. Endpoints may also occur further away from the section boundary for microtubules that naturally end there. Stitching consists of aligning sections, matching corresponding endpoints at two facing section boundaries, and connecting corresponding lines across multiple sections. See Figure 1 for an example that illustrates the complexity of the task and a solution achieved with our methods. Manual stitching of microtubule centerlines is labor intensive, and sometimes it is even impossible for a human to reliably decide how to align sections and how to connect the lines. Automated processing would be an attractive alternative, in particular, because the volumes reconstructed using electron tomography have been constantly increasing towards covering full cells (for example, Höög et al. [2, 3]), or large dense structures like centrosomes during cell division (for example, Müller-Reichert et al. [4], O'Toole et al. [5]). A purely manual segmentation of even larger structures, such as the massive meiotic spindle of *X. laevis* (see Brugués et al. [6], Loughlin et al. [7]), seems unrealistic.

Automatic stitching of microtubule endpoints from consecutive sections is challenging. The samples suffer from shearing and material loss at the boundary of sections during cutting and image acquisition. Samples shrink, warp and twist as a consequence of the electron exposure (see Figure S4 and also Luther [8]). Furthermore, automated and manual segmentation, too, will not always perfectly capture the endpoints (see Weber et al. [1]). Since microtubules, on the other hand, can be as close to each other as 25 nm (Lacomble et al. [9], Höög et al. [2,10], McDonald et al. [11], Ding et al. [12]), the correspondence of endpoints of close-by microtubules may be difficult to decide. A stitching method must handle the described deformations and be robust in the presence of noise.

To the best of our knowledge, the only software for stitching microtubule endpoints using the segmented centerlines is a semi-automatic tool integrated into the IMOD software (Kremer and Mastronarde [13]). Tomograms are roughly aligned using manually selected landmarks that are used to correct the deformation with a thin-plate spline model. Microtubules are traced afterwards in the aligned tomograms. Identified microtubules are then used to refine the deformation correction. This approach has been used successfully, for example, by McIntosh et al. [14], O'Toole et al. [5,15], and Höög et al. [16]. However, selecting matching microtubules manually is time consuming, tedious and, depending on the contrast of the tomogram and the presence of features to guide the user to potential matches, it can be infeasible.

An automated method could facilitate the stitching of microtubule centerlines over long distances in a reliable manner. Assuming that microtubule centerlines have been traced for a stack of electron tomograms of consecutive serial sections, two related problems need to be solved: 1) The sections need to be aligned, that is they need to be rotated and translated, and local distortions that are caused by tomogram acquisition need to be corrected. 2) Corresponding microtubules need to be matched and connected across section boundaries. If all centerlines and a perfect alignment were available, the matching should be obvious, because corresponding endpoints should be closer to each other than to any other endpoint. If a perfect endpoint matching was known, it could be used to compute an alignment such that the matched endpoints would be transformed onto each other. In practice, however, a perfect solution cannot be expected due to the limited data quality.

Solutions to the alignment and the matching problem have been studied before. A common method for finding an initial alignment of points is to identify a subset of matching points using transformation

invariant features and then compute the initial alignment from these pairs. Point pairs can be sampled, for example, with RANSAC (see Fischler and Bolles [17]) based on a local feature descriptor such as the sorted distance to neighboring points (see for example Lee et al. [18]). This method has been successfully applied in point registration in biology (Preibisch et al. [19]), image registration (Saalfeld et al. [20]), and to register lines (Yao et al. [21]). RANSAC-based methods have been demonstrated to perform well even in the presence of noise, outliers and strong deformation.

If only a few points need to be aligned, the problem can be formulated as finding cliques in a graph representation, called the distance compatibility graph (DCG). This method has been used for molecular shape analysis (for example Baum et al. [22]). It has also been applied for matching of neuron ends (for example Dercksen et al. [23]).

The most popular algorithm for the registration of points that are already coarsely aligned is the iterative closest points algorithm (reviewed in Rusinkiewicz and Levoy [24]). A probabilistic variant that formulates the problem in terms of a Gaussian mixture model was proposed by several authors (Rangarajan et al. [25], Wells [26]). The probabilistic variant has many advantages, including that the deviation of the assumed Gaussian mixture model can serve as a measure of uncertainty of the result; that the method naturally deals with outliers; and that the method is easy to implement. The approach has also been adapted for solving elastic deformation models (Jian and Vermuri [27], Myronenko and Song [28]).

Once the points are aligned, a common approach to finding unambiguously corresponding pairs is to compute a maximum weighted matching (MWM) on a bipartite graph (see Kuhn [29]). If a bipartite graph cannot capture enough prior information about the data, the matching can also be formulated in terms of a Markov random field (see Koller and Friedman [30]). This has received a lot of attention (for example, Caetano et al. [31], Sanghavi et al. [32]) and has been successfully applied to many applications, the closest to ours being the one described by Amat et al. [33], where the point matching computation is used to find a proper alignment of series of transmission electron microscopy images using the gold particles on top of the sample for alignment.

In the following, we present computational methods for the alignment and matching of microtubule centerlines that build upon the described work. See Figure 2 for an overview. 1) For the initial coarse alignment, we identify cliques in a distance compatibility graph. 2) For a fine alignment of the endpoints, we extend Myronenko and Song's work [28] by integrating line orientation. 3) For establishing correspondences between endpoints, we formulate a matching problem in terms of a Markov random field similar to Amat et al.'s [33]. Matchings are computed with belief propagation on a factor graph. Because belief propagation does not necessarily converge to a minimum in general, we propose a scheme to seek user input to resolve non-converging cases. With little, although carefully selected input, convergence is achieved for our application. We demonstrate the utility of the approach on electron tomograms of microtubules in *Caenorhabditis elegans* mitotic spindles in early embryos (*C. elegans*), *Xenopus laevis* oocyte meiotic spindles (*X. laevis*) and the sub-pellicular microtubule skeleton of *Trypanosoma brucei* (*T. brucei*). The combination of sample preparation and computational methods allows automated stitching of microtubule centerlines with less that 5% connections that disagree with an expert's opinion at each section boundary.

Abbreviations CPD: Coherent point drift. D: Disagreements between two point matchings. DCG: Distance compatibility graph. E-step and M-step: the two steps of the expectation-maximization algorithm. FN: False negatives. FP: False positives. MAP: Maximum a posteriori. MWM: Maximum weighted matching for point matching. PGM: Probabilistic graphical model for point matching. SVD: Singular value decomposition.

Materials and Methods

Sample preparation and tomography

Sample preparation Three different types of samples were used. Samples were prepared using well-established protocols. 1) Bipolar spindles were assembled from *X. laevis* egg extract using the previously published protocols Hannak and Heald [34] and Murray [35]. In brief, eggs collected from adult frogs were cleaned, de-jellied and centrifuged at 16,488 rcf for 12 min. 100 μl of the collected cytosolic fraction, which is naturally arrested in metaphase, was sent to interphase with the addition of 0.4 mM calcium and sperm nuclei. Once in interphase, equal volumes of the collected cytosolic fraction and the now interphasic fraction were mixed with 1 μl of Cy3-labelled tubulin to generate bipolar spindles. Animal use was approved by The Institutional Animal Care and Use Committee of The European Molecular Biology Laboratory (Permit Number: CA0555005). 2) As described in Müller-Reichert et al. [36], *C. elegans* (Bristol N2) expressing GFP-tagged tubulin were cut in half and individual single-cell embryos were extracted and sucked into capillary tubes (Leica) cut to 1 mm length. The capillary tubes were placed in 100-μm-deep membrane carriers (Leica) in M9 buffer containing 20% BSA. The cell cycle of embryos was followed by fluorescence miscroscopy until metaphase was reached. 3) As decribed in Höög et al. [37], *T. brucei* cells were grown at logarithmic phase in HMI-9 medium supplemented with 15% (v/v) HIFCS at 37°C. Cells were pelleted by gentle centrifugation (600g). A few microliters of pellet were transferred to membrane carriers.

The specimens (*X. laevis*, *C. elegans* embryos or *T. brucei* cells) were high-pressure frozen using a Leica EMPACT2 or HPM-010 high-pressure freezer, and freeze-substituted in a Leica AFS2. The substitution cocktail contained 2% uranyl acetate in acetone for *T. brucei*; 1% osmium tetroxide, 0.1% uranyl acetate in acetone for *C. elegans*; and 0.1% Tannic Acid, 0.2% glutaraldehyde, 2.5% water in acetone followed by 1% osmium tetroxide, 0.1% uranyl acetate in acetone 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 and collected on formvar-coated slot copper grids for EM. *T. brucei* samples were post-stained with 2% aqueous uranyl acetate (8 min) followed by Reynold's lead citrate [38] (3 min). *C. elegans* 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 (12 min).

Electron tomography 10 or 15 nm colloidal gold fiducial particles were deposited on the grids before imaging. Electron tomography tilt series were acquired in a Tecnai F30 electron microscope (FEI Company Ltd., Eindhoven, The Netherlands) operated at 300 kV at 1° tilt increments between −60 and +60 degrees, with the SerialEM software (Mastronarde [39]) using a Gatan US1000 2k camera or a FEI 4k Eagle camera. The pixel size was 1–2.5 nm. To minimize sample shrinkage during acquisition, the samples were pre-irradiated with a dose of 2000 electrons per square Angstrom or more, resulting in pre-shrinkage of the sample. During tomogram acquisition, the samples were submitted to a smaller or equivalent dose of electrons as during pre-irradiation. The tomograms were reconstructed and flattened using the IMOD software package as described in Kremer et al. [13]. Standard settings for plastic tomography were used for the fine alignment of 2D projections, including corrections for rotation, magnification changes due to defocus, distortions, and tilt angles, based on the tracking of gold fiducials on two surfaces of the grids. Due to the large area of interest, 1×2 to 1×4 frame montages were acquired for *T. brucei*; for *X. laevis*, 3×3 montages (11.4 μm×11.4 μm) were acquired 3 or 4 times per section, before being joined into large supermontages (11.4 μm×32 μm).

Microtubule tracing Microtubule centerlines were traced for each section individually from tomograms that had not been aligned between sections. For the *X. laevis* and *C. elegans* samples, centerlines were traced automatically as described by Weber et al. [1]. The supermontage data was binned prior to tracing. Automatically traced segments were manually validated and corrected using Amira [40]. The

segmentation for *T. brucei* was performed manually by an expert using the IMOD software (Kremer et al. [13]).

Computational methods

The input to the computational methods are centerlines from unaligned sections. The final result is a pairwise alignment of sections and a matching of endpoints that is applied to connect corresponding lines across sections.

The algorithms use a model for two facing section boundaries as illustrated in Figure 3. Section boundaries are modeled as parallel z-planes. The underlying assumptions are that sections can be cut reasonably parallel and that deformations during tomogram acquisition were minimized by a) coating the grids with carbon to minimize beam damage and sample distortion, b) pre-irradiating a wide area of the sample before tomogram acquisition to minimize local deformations during acquisition, c) by flattening the tomograms after reconstruction. Every section tomogram was visually inspected for flatness and corrected in IMOD or Amira if necessary. Top and bottom tomogram boundaries, therefore, were nearly parallel z-planes before microtubule tracing.

We assume that microtubules cross the section boundary as straight lines. This seems reasonable for microtubules that are perpendicular to the section boundaries, since such microtubules do not curve much over the thickness of a section (300–350 nm). The straightness assumption may be problematic, however, for microtubules that are parallel or nearly parallel to the section boundaries. In practice, the model should be reasonable if enough microtubules are sufficiently perpendicular. Results should be carefully evaluated, though, for regions were microtubules seem to be oriented mostly parallel to the section boundaries.

The endpoints are modeled as points that are located exactly at the section boundary. For microtubules that cross the section boundary, it seems reasonable to assume that they end exactly at the boundary even though real centerlines may contain some positional errors. Microtubules, however, may also naturally end within a section. In practice, the model should be reasonable if more microtubules cross the section boundary than end within a section. The algorithms include means to handle outliers. We simply treat all endpoints as if they ended at the boundary and leave it to the algorithms to handle endpoints without corresponding endpoint in the next section. *X. laevis* samples seem to meet the assumption well (the centerlines displayed in Figure 3 are a typical example). But also for the *C. elegans* samples, which contain many short lines close to the centrosome center, we have not observed problems in practice; the assumption seems to be sufficiently fulfilled.

The final result is computed in three main steps as illustrated in Figure 2. During initial alignment, each section pair is coarsely aligned using a linear transformation. During fine alignment, first a refined linear transformation is applied to each section pair followed by an elastic transformation to correct deformations. The matching identifies corresponding endpoints across the section boundaries, which is finally used to connect lines across multiple sections. We apply all steps in order they are described unless explicitly stated otherwise.

Notation Since we assume that all microtubule endpoints lie in a plane, their positions can be described by two-dimensional coordinates. The N endpoint positions $x_1, ..., x_N$ with $x \in \mathcal{R}^2$ are collectively denoted by X and the M endpoint positions $y_1, ..., y_M$ in the neighboring section by Y. The corresponding line orientations $\vec{x}_1, ..., \vec{x}_N$ with $\vec{x} \in \mathcal{R}^3, ||\vec{x}|| = 1$ are denoted by $\vec{X}$ and the orientations of the neighboring section $\vec{y}_1, ..., \vec{y}_M$ by $\vec{Y}$. $\mathcal{T}(y, \Theta)$ denotes a transformation with parameters Θ. For example, if $\mathcal{T}(y, \Theta) = sRy + t$ with scaling s, a rotation matrix R and a translation t, then $\Theta = (s, R, t)$. For the fine alignment, we will introduce a continuous random variable $\mathbf{x}$ and a discrete random variable $\mathbf{m}$ that has states $\{y_1, ..., y_M\}$. We will write $P(\mathbf{x})$ to refer to the probability density function of $\mathbf{x}$ and abbreviate the probability of $P(\mathbf{x} = x_i)$ by $p(x_i)$. Similarly we will write $P(\mathbf{m} = y_m) = p(y_m) = p(m)$ for the discrete

random variable. When we describe the Markov random field for endpoint matching, we will introduce one discrete random variable $\mathbf{x}_i$ for each endpoint.

Initial alignment

Initial alignment algorithm. For the initial alignment, a linear transformation of the two sets of endpoints X and Y is computed by finding cliques in a distance compatibility graph (DCG) (see Dercksen et al. [23]). Each clique of the DCG establishes a one-to-one correspondence between some points from X and Y, which can be used to compute an alignment. The method works by identifying similar spatial patterns in X and Y. The nodes of the DCG are pairs (x_i, y_k) of endpoints from X and Y. Two DCG nodes (x_i, y_k) and (x_j, y_l) are connected by an edge if $\big|\, ||x_i - x_j|| - ||y_k - y_l|| \,\big| < d$, where d is a threshold $\in R^+$. Such a graph is called a d-bounded DCG. In the worst case, the memory requirement for building the graph is $(NM)^2$, since there are NM nodes and each of these nodes can be connected to every other node. In practical applications, however, it is much smaller. Now consider a maximal connected subgraph (clique) in the DCG. Because all nodes in this subgraph are mutually connected, all distances $||x_i - x_j||$ of endpoints from X in the clique are close to the respective distances $||y_k - y_l||$ of the endpoints from Y. Therefore the endpoints x_i in the clique and the paired y_k are arranged in a similar spatial pattern. To find a good initial alignment, we search for large cliques in the DCG. This method is only successful if the distortions of the point sets are small.

For the alignment of microtubule endpoints, we can further restrict the number of edges in the DCG by taking into account the line orientations. Thus, we only connect two nodes by an edge in the DCG if $|\arccos(\vec{x}_i^T \vec{x}_j) - \arccos(\vec{y}_k^T \vec{y}_l)| < \alpha$. With this modification, we find cliques using the Bron-Kerbosch algorithm [41], similar to Dercksen et al. [23]. 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 contain fewer than $10,000$ nodes. In practice, the number of vertices in X and Y should be less than 100. This is often much fewer than the number of endpoints of microtubule centerlines in tomograms. We therefore use a heuristic to reduce the number of points: First, we sort all endpoints in descending order of angle of the line with the cutting plane of the section. Then, we pick the first 50 to 100 endpoints in each list. The microtubules that are most perpendicular to the plane of sections are expected to be easiest to match in the initial alignment. Furthermore, we only consider a clique if its size is at least 10 to 30% of the number of nodes in the smaller set. Also, we only consider the first 1000 cliques computed by the Bron-Kerbosch algorithm.

From the resulting endpoint correspondences, we compute an optimal (in a least-square sense) rigid alignment of the form $\mathcal{T}(y, \Theta) = Ry + t$ using singular value decomposition (SVD) as described by Umeyama [42] and Myronenko and Song [43]. To compute this alignment, we only take into account the endpoint positions. Line orientations could probably be incorporated by formulating the transformations with dual quaternions as described by Walker et al. [44]. The method described above, however, seems sufficient to compute a coarse initial alignment for sufficiently straight lines.

Fine alignment

The initial alignment considered only a subset of endpoints. We next describe how to compute a fine alignment from all endpoints for a linear and an elastic transformation model. The main text summarizes the methods. A detailed derivation of equations and the precise algorithms are given in the Supporting Information. Our fine alignment algorithms build upon the work of Myronenko and Song [28]. After briefly recapitulating their approach, we describe our extension to incorporate line orientations in the formulation. A basic understanding of probability distributions and statistical learning is assumed. For a practical introduction to statistics, see, for example, Dekking et al. [45], or Bishop et al. [46] for a more thorough treatment.

General methodology The general idea is to formulate a probabilistic model and find the transformation $\mathcal{T}$ that maximizes the likelihood of the data. The probability of a point $x \in X$ being a match for $y \in Y$ is modeled as a joint distribution of a D-dimensional continuous random variable $\mathbf{x}$ and a discrete random variable $\mathbf{m}$ with states $\{y_1, y_2, ..., y_M\}$

$$P(\mathbf{x}, \mathbf{m}) = P(\mathbf{m})P(\mathbf{x}|\mathbf{m}). \tag{1}$$

For two sets of two-dimensional points, $P(\mathbf{x}|\mathbf{m})$ will be a Gaussian $\mathcal{N}(\mathbf{x}; \mu, \sigma^2)$ located at the transformed y_m. $\mathcal{T}$ will be a transformation $\mathcal{T}(y_m, \Theta) = s\tilde{R}y_m + t$, where s is a uniform scaling factor, $\tilde{R}$ is a two-by-two rotation matrix, and t is a translation vector. Written out, $P(\mathbf{x}|\mathbf{m})$ then reads:

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

The prior $P(\mathbf{m})$ is uniformly distributed as $1/M$ since we have no measure of certainty of the ys and therefore assume that each is equally valid.

To find the optimal parameters $s, \tilde{R}, t$ and σ, Myronenko and Song use the expectation maximization algorithm. Two steps are performed iteratively to compute optimal transformation parameters (illustrated in Figure 4). First, the posterior $P(\mathbf{m}|\mathbf{x}) = \frac{P(\mathbf{x},\mathbf{m})}{P(\mathbf{x})}$ is computed in the expectation step (E-step). Here, $P(\mathbf{x}, \mathbf{m})$ is given by Equation 1 and $P(\mathbf{x})$ can be obtained from $P(\mathbf{x}, \mathbf{m})$ by marginalizing out $\mathbf{m}$. 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. Second, 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

$$Q = -\sum_{n=1}^{N}\sum_{m=1}^{M} p^{old}(m|x_n) \log(p(m)p(x_n|m)). \tag{3}$$

The first term, p^{old} is the posterior $P(\mathbf{m}|\mathbf{x})$ which was computed in the E-step and is kept fixed during optimization. Q is minimized with respect to the new transformation parameters Θ and the parameters of the distribution. In case $\mathbf{x}$ is distributed as a Gaussian and $\mathcal{T}$ is the transformation from above, we would minimize Q with respect to σ^2 and $(\tilde{R}, s, t)$. The detailed algorithm is given in Myronenko and Song [28].

The expectation maximization algorithm always converges to a local minimum (see Bishop et al. [46]). To adapt this algorithm for lines we 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.

Linear alignment For the linear alignment, we model the transformation of the line orientations as a rotation matrix $R \in \mathcal{R}^{3\times3}, \mathcal{T}(\vec{y}, \Theta) = R\vec{y}$. Assuming that all endpoints lie in a plane, we restrict the rotation to be around an axis that is perpendicular to the cutting plane, so that

$$\mathcal{T}(\vec{y}, \Theta) = R\vec{y} = \begin{pmatrix} \tilde{R} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{pmatrix} \vec{y}, \tag{4}$$

where $\tilde{R} \in \mathcal{R}^2$ is a two-dimensional rotation matrix. We model the transformations of the endpoint positions as $\mathcal{T}(y, \Theta) = s\tilde{R}y + t$, where $\tilde{R}$ is the same two-dimensional rotation matrix, $s \in \mathcal{R}$ is a uniform scaling parameter, and $t \in \mathcal{R}^2$ is a translation vector.

Algorithm for linear alignment from orientation. Our first linear alignment algorithm uses only the line orientations during expectation maximization, from which only the rotation $\tilde{R}$ can be directly computed.

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

$$P(\vec{\mathbf{x}}, \vec{\mathbf{m}}) = P(\vec{\mathbf{m}})P(\vec{\mathbf{x}}|\vec{\mathbf{m}}). \tag{5}$$

Because $\vec{\mathbf{x}}$ is periodic, we assume that $P(\vec{\mathbf{x}}|\vec{\mathbf{m}})$ is distributed as a Fisher-Mises distribution (see Mardia [47]):

$$P(\vec{\mathbf{x}}|\vec{\mathbf{m}}) = \mathcal{F}(\vec{\mathbf{x}}; \vec{y}_m, \kappa) = \frac{\kappa}{2\pi(e^{\kappa} - e^{-\kappa})} \exp(\kappa \mathcal{T}(\vec{y}_m, \Theta)^{\top} \vec{\mathbf{x}}). \tag{6}$$

The Fisher-Mises distribution is equivalent to a Gaussian distribution on a sphere (illustrated in Figure 5). The prior $P(\vec{\mathbf{m}})$ is uniformly distributed as $1/M$ as in the work of Myronenko and Song [28].

To obtain the optimal rotation in the M-step, we write down the expectation of the negative complete log-likelihood function (Equation 3) and minimize Q with respect to R and κ. We give details of this derivation in the Supporting Information. In brief, the optimal rotation can be obtained with a singular value decomposition (see Umeyama [42] and Myronenko and Song [43]). κ cannot be obtained in closed form, but it can be computed with Newton's method.

Even though we did not consider the endpoint positions during expectation maximization, we can apply the final rotation to the positions and compute a translation vector from all potential pairs weighted by the posterior (see Supporting Information). The complete algorithm is summarized in Figure S1.

Algorithm for linear alignment from position and orientation. Our second linear alignment algorithm considers both endpoint positions and line orientations and uses the full linear transformation model as introduced in Equation 4 and below. We define a joint distribution $P(\mathbf{x}, \mathbf{m}, \vec{\mathbf{x}}, \vec{\mathbf{m}})$ that factorizes as follows:

$$P(\mathbf{x}, \mathbf{m}, \vec{\mathbf{x}}, \vec{\mathbf{m}}) = P(\mathbf{x}|\mathbf{m})P(\vec{\mathbf{x}}|\vec{\mathbf{m}})P(\mathbf{m}, \vec{\mathbf{m}}). \tag{7}$$

$P(\mathbf{x}|\mathbf{m})$ and $P(\vec{\mathbf{x}}|\vec{\mathbf{m}})$ are defined by Equation 2 and 6 respectively. We define the prior $P(\mathbf{m}, \vec{\mathbf{m}})$ by

$$P(\mathbf{m}, \vec{\mathbf{m}}) = \frac{1}{M} \delta_{\mathrm{ind}(\mathbf{m}), \mathrm{ind}(\vec{\mathbf{m}})}, \tag{8}$$

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 0 otherwise. We chose this prior because we know the pairs $(y_j, \vec{y}_j)$, and pairs with mixed indices $(y_i, \vec{y}_j), i \neq j$ do not occur.

We can again write out Q for this distribution and the transformation model and then solve for the $s, \tilde{R}, t, \sigma^2$ and κ that minimize Q in the M-step. However, we do not have a closed-form solution for the optimal parameters. Instead, we minimize Q numerically. We give a derivation of the update equations and the first and second order derivatives necessary for the numerical optimization in the Supporting Information. For optimization, we use the library IPOpt [48]. The algorithm is summarized in Figure S2.

Elastic alignment A linear transformation cannot correct distortions, which usually appear in electron tomograms. To correct such distortions, we apply an elastic alignment.

Algorithm for elastic alignment. We model the transformation as individual translation vectors $\nu(y_k)$ that modify the positions such that $\mathcal{T}(y_k, \Theta) = y_k + \nu(y_k)$. The formulation closely follows Myronenko and Song [28]. They obtain the translation vectors by minimizing the regularized Q function with calculus of variations. The resulting vectors $\nu(y_k)$ are the product of an $M \times M$ smoothing matrix $\mathbf{G}$ and an $M \times 3$ matrix $\mathbf{W}$, that is $\mathcal{T}(y_k, \Theta) = y_k + \mathbf{G}(k, \cdot)\mathbf{W}$, where $\mathbf{G}(k, \cdot)$ denotes the kth row of $\mathbf{G}$. Here the transformation parameters Θ 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. Myronenko and Song call the algorithm for computing the optimal $\mathbf{W}$ the coherent point drift algorithm (CPD) for non-rigid point set registration.

Incorporating orientation into the described transformation model is not obvious, because the rotation that should be applied to the orientations is not apparent from the translation vectors $\nu(y_k)$. In practice,

however, we can assume that orientations are fixed, because we performed a linear alignment before. It is therefore reasonable to assume that the remaining elastic deformation contains negligible rotation. With this assumption, the orientations only influence the result via the posterior $P^{old}(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}})$. κ can be updated in each iteration again with Newton's method. See details in the Supporting Information. The algorithm is summarized in Figure S3.

We use the moving least squares algorithm as described by Schaefer et al. [49] to interpolate the deformation and apply it to the tomograms. The displacements vectors $\nu(y_k) = \mathbf{G}(k, \cdot)\mathbf{W}$ define the landmarks for moving least squares.

Endpoint matching using Markov random field

The previously defined distribution describe the probability that endpoints correspond. However, we cannot simply assign each endpoint in X to its most probable counterpart in Y, because assignments might conflict and the final pairs might not be unique. A common approach to finding unambiguously corresponding pairs is to compute a maximum weighted matching (MWM) on a bipartite graph (see Kuhn [29]). Furthermore, neighboring assignments might influence each other. For example, we would expect that the assignment of endpoints in Figure 6A is preferable to the assignment in Figure 6B, because remaining deformations of the data should be coherent in a neighborhood.

To find a matching of the endpoints, we use a Probabilistic Graphical Model (PGM) (see Koller and Friedman [30] for a comprehensive introduction) to model the influence of neighboring assignment on a pairing decision. Here, endpoints in X are represented by discrete random variables $\mathbf{x}_1, ..., \mathbf{x}_N$, each of which can have $M + 1$ states: either $y_1, ..., y_M$ to indicate the matching endpoint in Y or the placeholder y_0 to indicate that there is no match. Each $\mathbf{x}_i$ must be assigned with the constraint that two $\mathbf{x}_i, \mathbf{x}_j$ cannot be assigned to the same $y_1, ..., y_M$. Multiple assignments to the placeholder y_0 are explicitly allowed. Figure 6 illustrates possible states for a joint distribution of three variables.

To find a good assignment, we first define the joint distribution $P(\mathbf{x}_1, ..., \mathbf{x}_n)$ and then determine the joint assignment that yields the largest probability. $P(\mathbf{x}_1, ..., \mathbf{x}_n)$ is defined by a Gibbs distribution (Koller and Friedman [30])

$$P(\mathbf{x}_1, ..., \mathbf{x}_N) = \frac{1}{Z} \prod_{i=1}^{N} \Phi_o(\mathbf{x}_i) \prod_{i,j=1}^{N,N} \Phi_{oo}(\mathbf{x}_i, \mathbf{x}_j), \tag{9}$$

each state of which is a valid matching. Z ensures that $P(\mathbf{x}_1, ..., \mathbf{x}_N)$ is a valid probability distribution. The Φ's are called *factors*. Since the $\mathbf{x}_1, ..., \mathbf{x}_N$ are discrete random variables with values from a finite set, the factors are discrete tables with entries that can be though of as weights that represent our beliefs about a particular assignment. The *singleton factors* $\Phi_o(\mathbf{x}_i)$ contain information on how likely $\mathbf{x}_i$ and each $y_1, .., y_m$ or 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.

Factor values for matching microtubule endpoints We use three types of information to fill the factors Φ_o in Equation 9. First, we assume that microtubules are rather straight and, therefore, expect the angle $\mathrm{d}_\alpha(\vec{x}, \vec{y}) = \mathrm{acos}(\vec{x} \cdot -\vec{y})$ to be small. Second, we expect that the positions of matching endpoints are close and, therefore, compute the *direct distance* $\mathrm{d_c}(x, y) = ||x - y||$. $\mathrm{d_c}$ measures the distance on a plane, which we assume all endpoints to lie on (see Figure 7). Third, we expect that the straight extensions of corresponding lines should meet. To measure how close they get, we define the *projected distance* $\mathrm{d_p}(x, y)$ as the distance of y to the intersection of the line through x and a plane at y that is normal to $\vec{y}$ (see Figure 7). Each of the three d's is used in an exponential distribution with parameters λ_p, λ_c, λ_α, for example $\Phi_p(\mathbf{x}_i = y_k) = \lambda_p \exp(-\lambda_p \, \mathrm{d_p}(x_i, y_k))$ (see also Amat et al. [33]). For assignments

to the placeholder y_0, we use placeholder distances d_p^0, d_c^0, d_α^0. The individual factors are multiplied to define the singleton factors

$$\Phi_o(\mathbf{x}_i = y_k) = \Phi_p(\mathbf{x}_i = y_k)\Phi_\alpha(\mathbf{x}_i = y_k)\Phi_c(\mathbf{x}_i = y_k). \tag{10}$$

To account for joint assignments, we fill the pair factors as in Amat et al. [33] by considering the pair of displacement vectors $y_k - x_i$, $y_l - x_j$ induced by a joint assignment $\mathbf{x}_i = y_k, \mathbf{x}_j = y_l$. We define $\mathrm{d_s}(x_i, y_k, x_j, y_l) = ||(y_k - x_i) - (y_l - x_j)||$. If $\mathrm{d_s}$ is large, then the displacement vectors disagree. For example, $\mathrm{d_s}(x_1, y_2, x_2, y_1)$ in Figure 6B would be larger than $\mathrm{d_s}(x_1, y_1, x_2, y_2)$ in Figure 6C, because the connections cross and the displacement vectors are farther apart. We want to prevent such joint assignments and, therefore, penalize large values of $\mathrm{d_s}$ by filling Φ_{oo} with

$$\Phi_{oo}(\mathbf{x}_i = y_k, \mathbf{x}_j = y_l) = \lambda_s \exp(-\lambda_s \,\mathrm{d_s}(x_i, y_k, x_j, y_l)). \tag{11}$$

For assignments that involve the placeholder y_0, we use the placeholder distance d_s^0. Furthermore, we set entries to 0 if $k = l$ to account for the mutual exclusiveness constraint (see Koller and Friedman [30]). To reduce the number of factors that have non-zero entries, we introduce thresholds that restrict assignments to include only close-by points x, y, specifically $\mathrm{d_c}(x, y) < t_c$, $\mathrm{d_p}(x, y) < t_p$ 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.

Maximum a posterior assignment Each valid matching is a single joint assignment in the distribution $P(\mathbf{x}_1, ...\mathbf{x}_N)$. The goal now is to compute the maximum a posteriori (MAP) assignment $\mathrm{argmax}_{\mathbf{x}_1,...\mathbf{x}_N} P(\mathbf{x}_1, ...\mathbf{x}_N)$, the joint assignment with the greatest probability. Computing a MAP assignment is not generally feasible, because the problem is NP-hard.

We use an approximation technique called belief propagation on a graph representation called a factor graph (see Koller and Friedman [30], Amat et al. [33] and Kschischang et al. [50]). Briefly, each node in a factor graph represents one of the factors as depicted in Figure 8. Nodes are connected if their factors share a variable. Messages that express the belief about a MAP assignment of a variable are passed between the nodes in a particular order (red arrows and numbers in Figure 8). 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. For example, in Figure 8, the node representing the pair factor for $\mathbf{x}_1, \mathbf{x}_2$ uses message 1 to compute a belief about the assignment of $\mathbf{x}_2$ and passes this belief to the node representing the singleton factor for $\mathbf{x}_2$ in message 4. The message passing finishes if all messages agree. The MAP assignment can be read from the messages in the graph after convergence.

Unfortunately, convergence 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 and the algorithm will not converge. This oscillatory behavior is often a local problem of a network (see Koller and Friedman [30], 401 pp, 570 pp). To achieve convergence, we identify the endpoints that cause oscillation. We then point an expert to these endpoints and ask to provide a manual 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. We then run belief propagation again to check if all conflicts have been solved and convergence is achieved. If not, the expert is asked for more assignments.

To identify nodes that cause oscillation, we compute the 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 possible states of $\mathbf{x}_k$. We collect for each variable each incoming message for all factors that have the variable. For example, in Figure 8, 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 the disagreement for each pair of messages. 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 whose messages disagree most is then shown to the expert. We refer to these endpoints as *critical nodes*.

Parameters The PGM algorithm for point matching has 11 parameters: the four weights $\lambda_p, \lambda_c, \lambda_\alpha, \lambda_s$, the three thresholds t_p, t_c, t_α, and the four placeholder distances $d_p^0, d_c^0, d_\alpha^0, d_s^0$. We determine the thresholds and the placeholder distances from their corresponding λ's by choosing a single placeholder significance parameter r in the range $[0, 1]$. The placeholder distances are then computed such that the cumulative distribution function of the exponential distribution $1 - e^{-\lambda d^0}$ reaches $1 - r$. Thus $d_p^0 = -\log(r)/\lambda_p$ and equivalently for d_c^0, d_α^0, and d_s^0. We use the same values for the thresholds: $t_p = d_p^0, t_c = d_c^0, t_\alpha = d_\alpha^0$. Intuitively, r controls how much of the cumulative distribution function is ignored by ignoring assignments to real endpoints above a certain distance. We used $r = 1\%$ in all our experiments.

We determine reasonable choices for the weights $\lambda_p, \lambda_c, \lambda_\alpha, \lambda_s$ with a maximum-likelihood estimate using the ground truth. To do so, we consider Equation 9 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 directly related to the mean of the data, we can compute $\lambda_p^{-1} = \frac{1}{N} \sum_{i=1}^{N} d_p(x, y)$ and equivalently for $\lambda_c, \lambda_\alpha$, and λ_s. Figure S12 illustrates the choice of the parameters for our samples.

The parameter choice influences two aspects of the result. First, while manual input about assignments necessarily enforces convergence of belief propagation after a finite number of steps, we cannot know in advance how many assignments and iterations are needed. To measure it, we experimentally tested the procedure on the evaluation samples described. We iteratively ran belief propagation (5 passes) and assigned the endpoints that caused oscillation automatically by reading the assignments from the ground truth until belief propagation converged (see Results).

Second, the parameter choice might influence the quality of the result. We distinguish three types of 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 in the ground truth. 3) Disagreements (D): An endpoint was matched differently in the ground truth and the automatic assignment. We also measure how many pairs were assigned correctly, that is, the true positives (TP). To estimate the sensitivity of the parameters $\lambda_p, \lambda_c, \lambda_\alpha, \lambda_s$, we varied their value around the maximum-likelihood estimate. For each value, we ran belief propagation and assured convergence using the ground truths as described before. We measured the number of iterations, the number of manual assignments, and the number of FN, FP and D resulting from the computed matching. Furthermore, to get an estimate on the error rate, we provide the precision P=$\frac{TP}{N_A}$, recall R=$\frac{TP}{N_M}$, and fraction of disagreeing matchings DIS=$\frac{D}{N_M}$. N_M here is the number of pairs in the ground truth and N_A the number of pairs in the automatic matching (see Results).

Implementation We use libDai [51] to run belief propagation on the network with the scheduling proposed by Elidan et al. [52]. The graphical user interface has been implemented in Amira [40].

Ground Truth for Evaluation

To prepare a ground truth, we applied the described computational methods. The programmer assigned evidence as she saw fit to the PGM point matching. The result was then verified and corrected by experts in biology using a graphical user interface (see Figure S5). The user interface supports inspection of the line geometry in a perspective view and in a separate view together with an oblique image slice. Both views can be interactively changed at any time. Different sections are indicated by colors. To support the verification process, the user interface automatically navigates to endpoints and restricts the perspective view to show only neighboring lines and endpoints. The experts made decisions primarily based on such closeup views. They inspected the three-dimensional situation by interactively changing the viewing

direction to verify that connections of neighboring lines were consistent. The experts felt that they could make reasonable decisions. They systematically verified all endpoints close to the section boundaries and corrected wrong connections or added missing connections. It was sometimes helpful to verify lines also in an overview that shows larger parts of a section. The experts did not add new lines that seemed to be completely missing in neighboring sections. We accepted missing lines, because the primary purpose of the evaluation was to determine whether the result of automatic alignment and matching agrees with an expert's opinion, and some lines are expected to be missing in tracings from individual sections.

Results

To understand the performance of the computational methods, we tested them on samples of different characteristics. We used more than 50 pairs of sections from *C. elegans* mitotic spindles for general testing. See Figure 1 for an example of a stitched stack of 20 sections. The *C. elegans* samples are relatively small in size (the centrosome is less than 5 μm in diameter). Microtubules extend from a centrosomal region that can be covered with a single frame tomogram. The samples contain approximately 1500 lines per section. Endpoints are spread in a ring-like structure around the microtubule organizing center and orientations are distributed homogeneously (Figure 9 top). We limited detailed testing to three pairs of consecutive sections, for which we created a ground truth. Two experts independently verified and corrected the computed matching for the whole section boundaries. One of the ground truth samples is displayed in Figure S6.

We also used a stack of three sections of a *X. laevis* meiotic spindle. The sample is larger, and tomogram montaging was necessary. Each section contains approximately 3500 lines. Finding a transformation is hard due to deformations. Establishing endpoint correspondence is difficult, because microtubules are organized in dense bundles. Figure 9 (middle) depicts endpoint positions and line orientations for a small area of these sections. Line orientations are clustered. Unlike in the *C. elegans* centrosome, most microtubules in the *X. laevis* spindle have a similar orientation. Since the sections are huge, we created a ground truth only for a subregion. Manual verification and correction was preformed on a region that contains approximately 1000 lines per section (see Figure S7). The subregion covers a relevant part of the tomogram and should contain a variety of typical configurations, which were indeed observed by the expert during verification. We did not, however, perform additional tests that the region is representative.

Furthermore, we used a sample of the sub-pellicular microtubule array of *T. brucei*, which is a sheet of parallel microtubules underlying the plasma membrane (reviewed in Gull [53], Farr and Gull [54]). Figures 9 (bottom) depicts endpoint positions and line orientations of two consecutive sections. Tomogram montaging was used. Each section contains 200–400 microtubules. This sample is challenging, because microtubules are arranged in sheets. No ground truth was prepared for the *T. brucei* data.

The running times of all algorithms were below 2 minutes per ground truth section pair (single threaded optimized code, CPU AMD Opteron 6174).

Initial alignment

Our initial alignment algorithm worked reliably for *C. elegans* tomograms and reasonably well for *X. laevis* tomograms, but it failed on *T. brucei* tomograms. A reasonable initial alignment was found for more than 50 pairs *C. elegans* sections. We used 50 endpoints in each section to build the DCG. In approximately 10% of the cases, we had to scale one of the two sections between -5% and $+5\%$ to obtain a reasonable result. We tried several uniform global scalings until we found a reasonable result. For four *X. laevis* samples (each contained approximately 3500 lines), we had to pick at least 100 endpoints from each section, restrict the size of the cliques in the DCG to 10 endpoints, and test different scalings to obtain a rough alignment. For *T. brucei*, we rotated the sections to simulate completely unaligned tomograms (Figure 10, top left). The algorithm failed to compute an initial transformation. No matching points

could be identified by comparing angles of line orientations. The reason probably is that all lines are organized in a regular pattern more or less in parallel with a fixed spacing (Lacomble et al. [9]).

Fine alignment

Robustness of linear alignment We tested robustness of the linear alignment and found that our algorithm for linear alignment from orientation (Figure S1) and our algorithm for linear alignment from position and orientation (Figure S2) seem to be a bit more robust than the original rigid point set registration algorithm by Myronenko and Song (Figure 2 in [28]). To test robustness, we rotated one section of a pair of correctly aligned sections for several samples over a range of 360° in 5° steps and tested whether the three algorithms were able to rediscover the original rotation. We used two samples from the *C. elegans* ground truth and two randomly chosen subregions from the *X. laevis* samples. Results are summarized in Figure 13. All algorithms found correct alignments for small rotations. The algorithm for linear alignment from orientation found the original rotation in more cases than the other algorithms. Upon convergence, σ^2 was $8.68 \cdot 10^{-5} \pm 1.25 \cdot 10^{-4}$ (mean ± standard deviation) when the rotation angle was correctly computed and $6.92 \cdot 10^{-2} \pm 7.09 \cdot 10^{-2}$ when the angle was incorrect. κ was 123 ± 16.4 for correct and 22.7 ± 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* sample (Figure 10) and observed that satisfactory results could only be achieved by applying the linear alignment in two steps. The original rigid point set registration algorithm by Myronenko and Song (Figure 2 in [28]) and our algorithm for linear alignment from position and orientation (Figure S2) both failed to compute a correct result from the initial alignment (Figure 10, top right). A combination of the following two steps, however, yielded a good result: By using our algorithm for linear alignment from orientation (Figure S1) first, a reasonable rotation could be computed (Figure 10, bottom left). In the second step, a refined alignment could then be computed (Figure 10, bottom right) with our algorithm for linear alignment from position and orientation (Figure S2).

Displacements during elastic alignment The original CPD algorithm (Non-rigid point set registration algorithm, Figure 4 in [28]) and our algorithm for elastic alignment (Figure S3) compute similar displacements. We compared the two algorithms by measuring the distances of the manually connected endpoints before and after applying the transformation to the *X. laevis* ground truth. Figure 11 shows the histograms of the distances. Both methods performed similarly. The original CPD, however, needed roughly 1.5 as many iterations. We measured similar factors for other samples.

Limitations The algorithm for elastic alignment yielded a reasonable result for most parts of the *T. brucei* sample. However, some parts were not aligned properly, see Figure S8. Here, the segmented lines were often too long or too short, and the assumption that point displacements are coherent in a small neighborhood was violated. The proposed deformation model failed to handle such cases.

Precisely locating the relevant part of a sample under an electron microscope is difficult, and serial tomograms, therefore, might be shifted to each other. If the shift is large, only a small portion of the lines in one section has a corresponding counterpart in the next section. To test whether the algorithms handles small overlaps, we cut two regions with little overlap out of a *X. laevis* (Figure S9) and applied the alignment algorithms. None of the algorithms yielded a reasonable result.

Endpoint matching

Parameters of the probabilistic graphical model The described maximum-likelihood estimate delivers reasonable parameters, and matching results are relatively insensitive to parameter variations. We used the ground truth of the *X. laevis* sample to estimate parameters: After linear alignment but

before elastic alignment, parameters were estimated as $\lambda_c^{-1} = 771\,\text{Å}$, $\lambda_p^{-1} = 577\,\text{Å}$, $\lambda_s^{-1} = 297\,\text{Å}$, and $\lambda_\alpha^{-1} = 5.8\,°$. After applying the algorithm for elastic alignment, parameters were estimated as $\lambda_c^{-1} = 243\,\text{Å}$, $\lambda_p^{-1} = 170\,\text{Å}$, $\lambda_s^{-1} = 293\,\text{Å}$, and $\lambda_\alpha^{-1} = 5.8\,°$ (see Figure S12). The smaller values for λ_c^{-1} and λ_p^{-1} confirm that the elastic alignment moved corresponding endpoints closer. To fine-tune the parameters and test their sensitivity, we varied each parameter separately around its maximum-likelihood estimate and measured the performance of the PGM matching on the *X. laevis* sample to which the algorithm for elastic alignment had been applied (see Figure 12). The performance was stable over a wide parameter range. We also estimated the parameters from the *C. elegans* samples to which the algorithm for elastic alignment had been applied: λ_c^{-1} was in the interval $(178\,\text{Å}, 300\,\text{Å})$, λ_p^{-1} was in the interval $(105\,\text{Å}, 121\,\text{Å})$, λ_s^{-1} was in the interval $(155\,\text{Å}, 247\,\text{Å})$, and λ_α^{-1} was in the interval $(6.7\,°, 8.1\,°)$.

We used the maximum-likelihood estimates that were computed from the *X. laevis* sample after elastic alignment for further experiments, except for the pair shift parameter λ_s^{-1}, which we set to 150 Å, because fewer manual corrections were required at this value (see Figure 12). We saw no reason to choose different parameters for individual specimens, because the matching was stable over a wide parameter range; the estimates from the *C. elegans* samples were in a similar range as the estimates from *X. laevis*; and the matching quality later turned out to be good for *C. elegans*.

Quality of matching We compared the result of the PGM matching with the ground truth samples. We used a placeholder significance of $r = 1\%$ in all experiments. Figure S10 displays typical differences that we observed for the *X. laevis* sample. When the PGM matching disagreed with the ground truth, the PGM's choice often seemed reasonable, too. Some situation with disagreement seemed inherently difficult to decide. In some situations, this might be caused by lines that are missing in one section. Figure S11 displays typical differences that we observed for the *C. elegans* samples. We observed far fewer disagreements than for the *X. laevis* sample. The reason probably is that lines are oriented more arbitrarily. They do not form bundles, and ambiguities seem less likely. We observed a few additional or missing lines in the PGM matching.

We found that the result of the PGM matching clearly outperforms an MWM for the three sections of the *X. laevis* spindle. Table 1 contains a quantitative comparison. We first set the parameters to their maximum-likelihood estimates from above and computed the matching for the sample to which the algorithm for elastic alignment had been applied (Table 1, configuration X1). To examine the effect of parameter variations, we ran the matching again with parameters set to twice their maximum-likelihood estimate (Table 1, configuration X2). To test stability of the PGM with respect to alignment quality, we ran the matching on the *X. laevis* sample to which the algorithm for linear alignment from position and orientation had been applied but not the algorithm for elastic alignment. We used the maximum-likelihood parameters as estimated above for this situation: $\lambda_c^{-1} = 771\,\text{Å}, \lambda_p^{-1} = 577\,\text{Å}$ and $\lambda_\alpha^{-1} = 5.8\,°$ except for the pairwise shift λ_s^{-1}, which we again set to 150 (Table 1, configuration X3). The results indicate that the PGM approach is unaffected by parameter variations and imperfect alignment in terms of precision (0.96), recall (0.95) and fraction of disagreeing matches (0.04); see Methods above for definition of the quality measures. MWM performed worse, with precision and recall dropping below 0.9 for parameters at twice the maximum-likelihood estimate and below 0.65 without elastic alignment. In all cases, only a small fraction of endpoints ($< 3\%$) had to be assigned manually to achieve convergence. It took an expert 1 h per 100 nodes to provide assignments. Checking unconnected lines after the final matching took 1h per 500 endpoints.

On *C. elegans* samples, the PGM and the MWM delivered comparable results. We tested on the six *C. elegans* ground truths (3 datasets, 2 experts). The bottom part of Table 1 displays the mean of the performance measures. Both approaches performed equally well (precision 0.97, recall 0.96). We did not measure a significant difference between experts. For the PGM, all networks always converged without user input.

Limitations We tested the PGM matching on the *T. brucei* sample to which the algorithm for elastic alignment had been applied. We found that it is difficult to assign the critical nodes due to the regular pattern formed by the microtubules. Figure S8 shows examples of configurations for which the algorithm requested assignments. We failed to decide whether lines were missing and which endpoints should be matched.

Discussion

We presented computational methods for automated stitching of filaments across serial sections. Our tests on microtubule centerlines indicate good agreement of the automated results with experts' opinions for spindle samples (approximately 5% disagreeing connections per section boundary), which suggests that the proposed computational methods can be used in practice to accelerate an expert analysis. It is difficult, however, to conclude how close the results are to the true physical reality. The ground truth that we used for measuring the accuracy was created by correcting the output of the proposed computational methods. The experts might have been influenced by the output and might have decided more similar to the computational methods compared to what they would have decided if they had started from scratch. Our measurements of accuracy might, therefore, be biased in favor of our approach. On the other hand, when the computational methods disagreed with the ground truth, both choices often seemed reasonable, which suggests that some connections might be inherently difficult to decide. We also observed situations in which lines seemed to be completely missing in one section, so that no reasonable connection could be made. Since some missing lines are expected (from our experience, 4% after automatic tracing, see Weber et al. [1]), we did not correct them in our evaluation. For the microtubule arrays in *T. brucei*, the methods failed to yield satisfactory results, but experts also found it difficult to decide locally how to connect lines. We think that the proposed methods will be useful for certain experiments, like the analysis of spindle microtubules, while they might not be immediately applicable in other experiments. A good indicator whether the methods will work is probably whether an expert is able to confidently decide locally how to connect microtubule centerlines across section boundaries.

Using line orientation for the alignment has several advantages. In the experiments on robustness of the linear alignment, our algorithm for linear alignment from orientation succeeded in more cases than the other algorithms. This algorithm furthermore was the only algorithm that succeeded in computing an initial alignment for the *T. brucei* sample. Both results suggest that including orientation in the formulation stabilizes results. Orientation probably helps avoiding local extrema during optimization that would arise if only endpoint positions were used. This seems reasonable if there are preferred orientations that clearly indicated a certain alignment, such as in the *X. laevis* and *T. brucei* samples (see Figure 9). But orientation might help, because it may contain different information than position in general. The additional information might be the reason that using orientation accelerated convergence of expectation maximization for our algorithm for elastic alignment by a factor of 1.5 compared to the non-rigid point set registration algorithm by Myronenko and Song (Figure 4 in [28]). Finally, line orientation allows an estimation of the quality of the result by inspecting the concentration parameter κ in addition to σ^2. A large κ and a small σ^2 were clear indicators that the alignment succeeded. The ability to assess the result in this way without ground truth can be valuable in practice.

Using orientation for stitching has received little attention in the literature as of yet. Our approach is similar to the approach by Hogrebe et al. [55] and Dercksen et al. [23], who use segmented neuron centerlines to compute an alignment for serial sections of neurons imaged with confocal microscopy. The major difference of our approach is that it uses orientation and probabilistic methods. We think that the latter is essential here, because centerlines traced in electron tomograms are subject to noise and artifacts, and modeling uncertainty might be key to achieving reliable results.

The probabilistic approach to endpoint matching seems important. The PGM matching clearly outperformed a maximum weighted matching (MWM) if only a linear alignment was applied but no elastic

alignment (Table 1, X3). Furthermore, the matching results were robust to parameter variations (Table 1, X2 and Figure 12). Both results suggest that a PGM is more reliable than a MWM and therefore the preferred approach in practice, although the performance of the MWM and the PGM matching were similar if the algorithm for elastic alignment had been applied and parameters were well chosen (Table 1, *X. laevis* X1 and *C. elegans*). For the *C. elegans* samples, the reason might be that orientation already limits the possible candidates and few ambiguities remain. The primary reason for the good performance of the PGM is probably that the pair factors enforce a coherent shift of neighboring assignments. Another reason might be that we seek user input for unclear situations. Manually assigned endpoints obviously agree with the expert's opinion in the final result. They probably also stabilize the decision on neighboring connections. Assigning endpoints manually is tedious, but in all cases, manual assignment of less than 3% of the endpoints was sufficient to achieve convergence. This seems to be an acceptable effort considering the quality of the result.

The PGM matching seems relatively robust. The maximum-likelihood estimates for the parameters were in a similar range for *X. laevis* and the *C. elegans* samples to which the algorithm for elastic alignment had been applied. The maximum-likelihood estimates for the direct and projected distances, however, were larger before elastic alignment. This is expected, because a successful alignment moves points closer and reduces the maximum-likelihood estimate, which is the mean distance between corresponding points. For similar alignment quality, similar differences in position and orientation of corresponding pairs can be expected even for different samples. The matching results were relatively unaffected by variations of the PGM parameters in our experiments on the *X. laevis* samples. Together this suggests that the same PGM parameters can probably be safely used for different samples if the alignment quality is known to be similar. But different parameters should probably be used for different experimental conditions that may introduce different errors.

The computational methods all rely on a model for facing section boundaries that makes certain assumptions about the tomograms (see details in section on computational methods): Boundaries are modeled as parallel z-planes, assuming that a skilled operator can conduct experiments such that section boundaries are relatively flat and parallel. Microtubules are modeled as straight lines, assuming that many microtubules are sufficiently perpendicular to the section boundary. Endpoints are modeled as points that are located exactly at the section boundary, assuming that more microtubules cross the section boundary than end within a section. The results suggest that the model is reasonable for the samples we used. For different conditions, however, it might be inappropriate and results should be carefully evaluated. Such conditions could be microtubules that are oriented mostly parallel to the section boundaries or many short microtubules that are likely to end within sections.

Despite the encouraging results, the limited accuracy should be considered in a subsequent analysis. Stitching errors may, in particular, bias statistics on length and number of microtubules. In our experiments, 95% of the computed connections agreed with an expert's opinion, but 4% (*X. laevis*) and 1% (*C. elegans*) connections disagreed (Table 1). Even if disagreements counterbalanced a bit, 2% to 4% of the connections might be completely missing. This error will add up for microtubules that traverse several sections. For example, the number of microtubules might be overestimated by 80% in a stack of 20 sections in the worst case. In practice, unconnected lines should always be verified and the impact of the observed errors should be estimated for a specific analysis. Such a manual verification takes time (one hour per 500 corrections in our experience). Yet, it is still substantially less work than connecting all lines completely manually. Another benefit of a manual verification is that other errors such as line tracing errors may also be detected (from our experience 4% of the lines may be missing after automatic tracing, see Weber et al. [1]).

A potential limitation of our current approach is that the initial alignment based on the DCG might not scale for a larger number of endpoints. Although the method was successful on all *C. elegans* and *X. laevis* examples, we believe that it might be too slow for samples with more than 4000 endpoints per section. If the global structure of the sample is obvious, a simple approach for initial alignment might

work, such as aligning the center and the eigenvectors of the mass distribution. If global structure is not apparent, a feature descriptor approach like Preibisch et al. [19] might be a suitable solution; or the initial alignment could be completely skipped and replaced with the algorithm for linear alignment from orientation. The algorithm could be applied from several starting points and the best solution could be automatically chosen by inspecting κ and σ^2.

The combination of methods that we propose might also be useful for stitching other filamentous structures. However, one major assumption of the algorithms is that lines are rather straight, which might not be the case for other applications. For curved lines, our approach could perhaps be improved by incorporating a measure of higher order line properties such as curvature. Even if lines are straight, complications might arise. For example, we believe that computing a matching for the sheets of parallel microtubules in *T. brucei* would require a different approach. Establishing correspondences is particularly challenging due to the periodicity in the sub-pellicular microtubule array.

An alignment of tomograms based on microtubules might be more accurate than an alignment based on a few manually selected landmarks. Applying the proposed methods may be useful even for samples that would otherwise be aligned manually. If tomograms contain microtubules in all relevant parts, it might make sense to segment microtubules using automatic tracing [1] and apply the proposed alignment algorithms even if microtubules are not the focus of that study.

In summary, we developed a tool to robustly stitch segmented microtubule centerlines across the gap between serial electron tomograms. We believe that our software will greatly facilitate research on microtubule organization in cell biology. Up to now, the analysis of microtubules in electron tomograms has been mostly based on observations of 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*. The approach we introduced here should allow the analysis of microtubule centerlines over long distances across serial electron tomograms for certain structures, such as the spindle apparatus.

Availability

Data to reproduce key results are available from the digital repository Dryad (DOI 10.5061/dryad.v8j20). At http://www.zib.de/en/visual/software/microtubulestitching.html, a binary version of software is available for download. It implements the methods as an extension package to the visualization software Amira (Windows and Linux, both 64-bit).

Acknowledgments

JLH's research was funded by a Sir Henry Wellcome postdoctoral fellowship. JLH's imaging was performed in The Boulder Lab for 3D Electron Microscopy supported by NIH-NCRR, Grant P41-RR000592 to Andreas Hoenger. We thank Jan Ellenberg and Francois Nédélec for mentorship of EMT. We thank Jana Meissner, Chris Weiss and Lisa Gottschalk for their technical support and expertise in *C. elegans* embryos. We thank Keith Gull and Richard McIntosh for mentorship of JLH; work in Keith Gull's laboratory was supported by the Wellcome Trust.

References

1. 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.

2. 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.

3. Höög JL, Antony C (2007) Whole-cell investigation of microtubule cytoskeleton architecture by electron tomography. Methods Cell Biol 79: 145-67.

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

5. 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.

6. Brugués J, Nuzzo V, Mazur E, Needleman DJ (2012) Nucleation and transport organize microtubules in metaphase spindles. Cell 149: 554–64.

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

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

9. 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.

10. Höög JL, Huisman SM, Seb-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.

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

12. 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.

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

14. 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.

15. 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.

16. Höög JL, Lacomble S, O'Toole ET, Hoenger A, McIntosh JR, et al. (2014) Modes of flagellar assembly in Chlamydomonas reinhardtii and Trypanosoma brucei. Elife 3: e01479.

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

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

19. 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.

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

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

22. 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.

23. 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.

24. 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.

25. 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.

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

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

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

29. Kuhn HW (1955) The Hungarian method for the assignment problem. Naval Research Logistic Quarterly 2: 83–97.

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

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

32. 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.

33. 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.

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

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

36. 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.

37. 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.

38. Reynolds ES (1963) The use of lead citrate at high pH as an electron-opaque stain in electron microscopy. J Cell Biol 17: 208–12.

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

40. 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.

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

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

43. Myronenko A, Song XB (2009) On the closed-form solution of the rotation matrix arising in computer vision problems. arXiv 0904.1613.

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

45. Dekking F, Kraaikamp C, Lopuhaä H, Meester L (2010) A Modern Introduction to Probability and Statistics: Understanding Why and How. Springer Texts in Statistics. Springer.

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

47. 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.

48. 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.

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

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

51. 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.

52. 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.

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

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

55. 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.

Figures

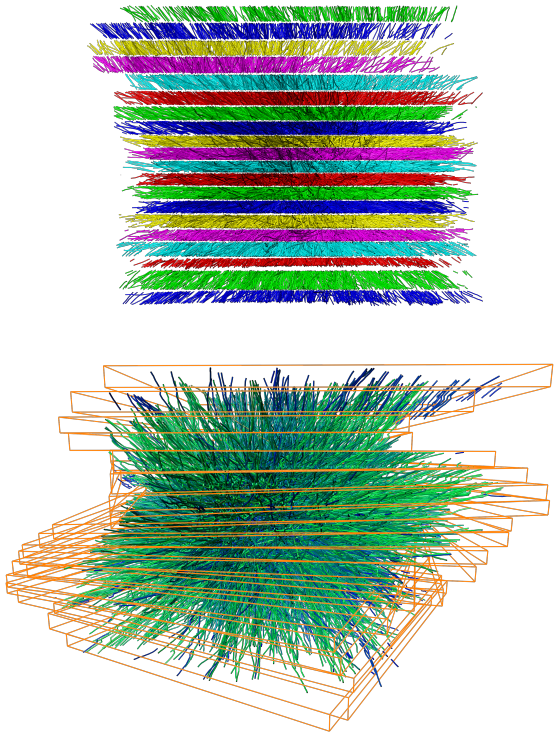

Figure 1. Starting point and output of our computational methods. Results are displayed for a stack of 300 nm thick and 5 μm wide sections of microtubule centerlines in the mitotic spindle of a *C. elegans* early embryo. Each section contains approximately 1500 microtubules, which were traced in electron tomograms. Top: Side view of unaligned centerlines before stitching. Colors indicate different sections. Bottom: Perspective view of the stitched microtubules after applying our algorithms. Color indicates length of the stitched microtubules: greenish lines are longer; bluish lines are shorter. The orange bounding boxes help getting an impression of the alignment transformation. The boxes were oriented parallel to the main coordinate axes before alignment. Some lines do not have a continuation in the next section (for example bluish lines at top right), because the area covered by the tomograms varied from section to section.

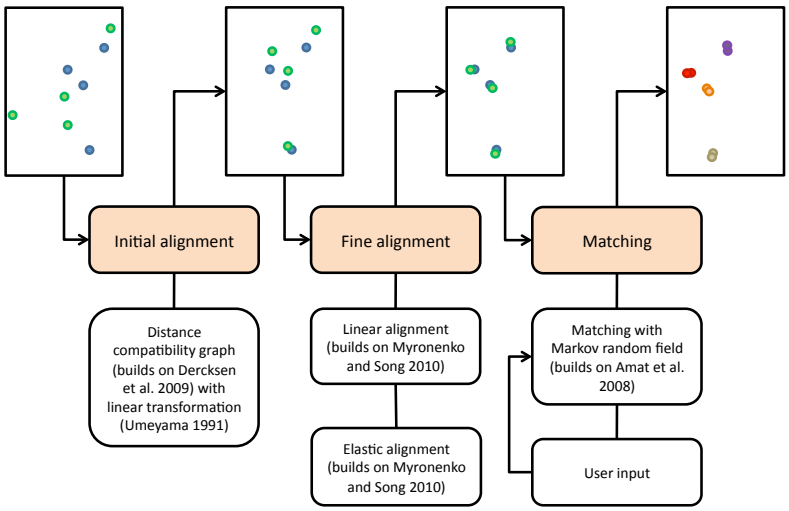

Figure 2. Computational methods. Top and middle: Illustration of two neighboring sections (view from top) and main processing steps. Bottom: algorithmic details and references. Endpoints are illustrated in blue and green. The initial alignment computes a coarse, linear alignment from a subset of endpoints. The fine alignment moves endpoints closer in two steps: a linear alignment followed by an elastic alignment. The matching determines pairs of corresponding endpoints (indicated in different colors), which are finally connected across sections (not illustrated).

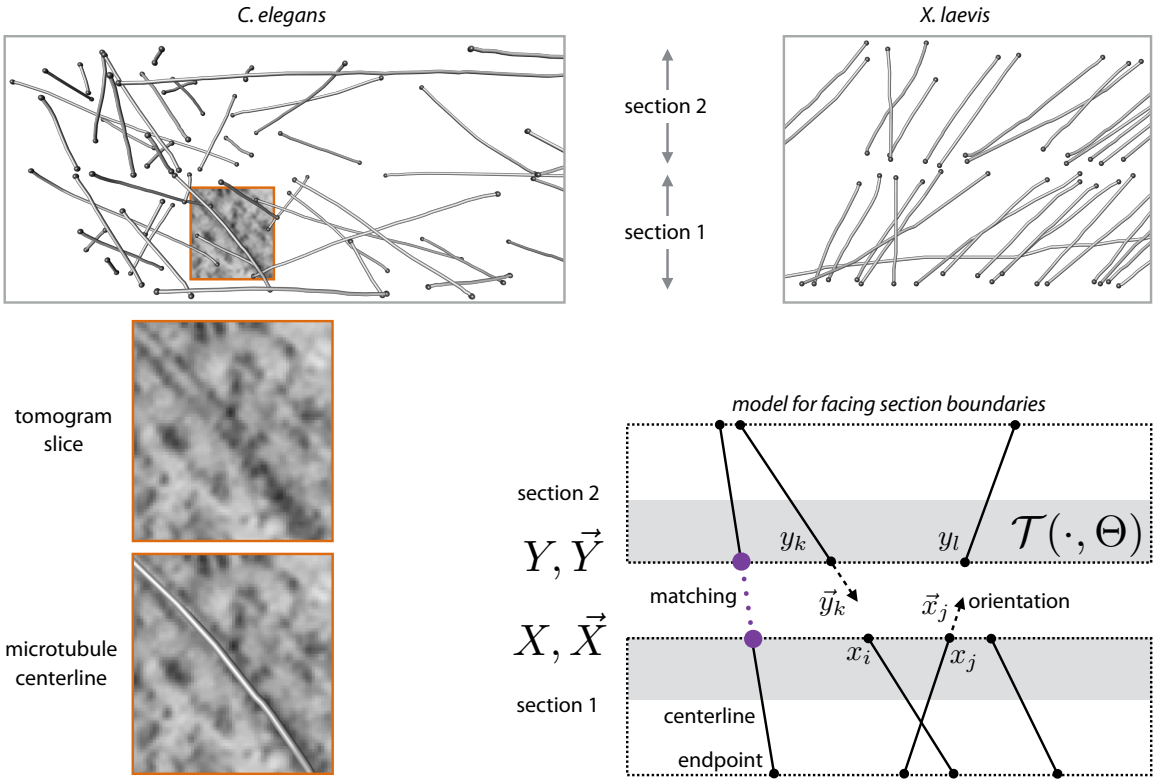

Figure 3. Model for section boundary. Top: View from side at traced microtubule centerlines for two small regions from two neighboring sections of a *C. elegans* sample and a *X. laevis* sample. For the purpose of illustration, the sections have been aligned. Many microtubules are regularly oriented in parallel bundles in the *X. laevis* sample. Microtubules are less regularly oriented in the *C. elegans* sample. Bottom left: Part of a centerline together with the corresponding tomogram slice. Bottom right: Model for boundary between two neighboring sections. Microtubule centerlines are modeled as straight lines with a single orientation (illustrated as dashed arrows). Endpoints close to a boundary (gray area) are described by two-dimensional coordinates as if they were located in a single plane. The transformation $\mathcal{T}$ acts on one section boundary (the y's in section 2). A matching consists of pairs of corresponding endpoints (one pair indicated as dotted line). The goal is to find a transformation and a matching such that corresponding centerlines can be connected across several sections. See main text for detailed notation.

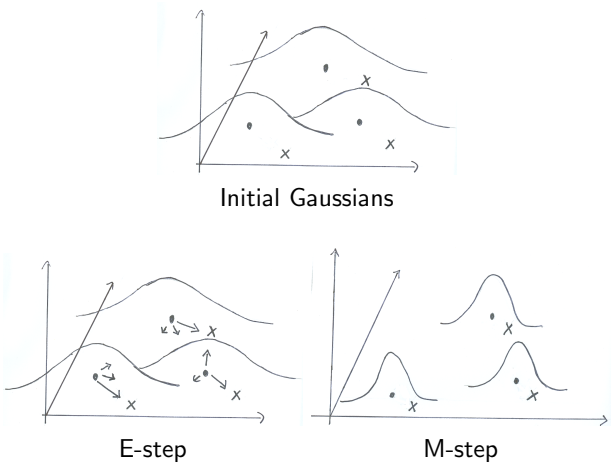

Figure 4. Illustration of the expectation maximization algorithm for a Gaussian mixture model. Dots represent the centroids of the Gaussians which are located at $\mathcal{T}(y_1, \Theta), ..., \mathcal{T}(y_M, \Theta)$. Crosses represent the points in X. E-step: Arrows indicate values of the posterior for each point in Y. M-step: Steps towards convergence move the points closer and reduce the width of the Gaussians. Upon convergence, corresponding points should be close to each other and the Gaussian should be sharply peaked.

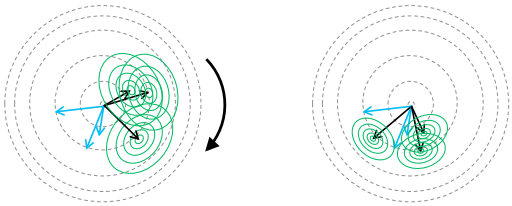

Figure 5. Illustration of the expectation maximization algorithm for periodic variables. The dashed circles illustrate the unit sphere. Arrows indicate the orientation unit vectors projected onto the plane (blue: $\vec{x}$, black: $\vec{y}$). The green circles depict values of the Fisher-Mises distribution with centers located at the arrow tips. In the M-step, a rotation around the center is computed that moves the arrow tips closer. Simultaneously, the concentration parameter κ is updated and the width of the Fisher-Mises distribution is reduced. Left: before M-step. Right: after M-step.

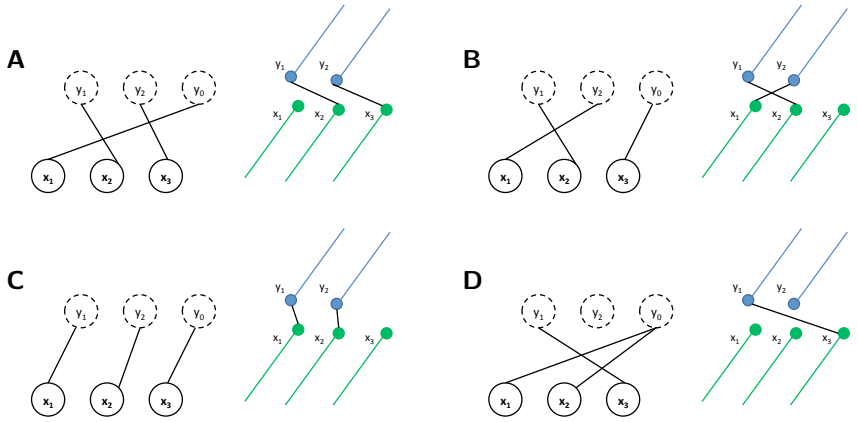

Figure 6. Examples of matching configurations. Centerlines from two consecutive sections are depicted in blue and green. The random variables $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are illustrated on the left of each panel. Connections to y_1 and y_2 indicate assignments to endpoints in the neighboring section. Connections to y_0 indicate assignments to the placeholder. A and C: consistent assignments. C should be preferred because the corresponding points are closer. B and D: assignments that should be penalized.

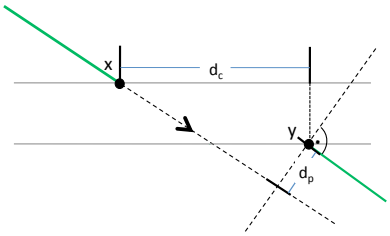

Figure 7. Endpoint distances. $\mathrm{d_c}(x, y)$ is the direct distance, which is computed as the horizontal distance between x and y. $\mathrm{d_p}(x, y)$ is the projected distance, which is computed by projecting x along the line orientation at x onto a plane through y that is perpendicular to the orientation at y.

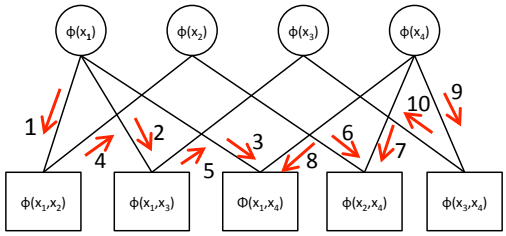

Figure 8. Illustration of a factor graph with message passing. Circles depict singleton factors. Squares depict pair factors. Red arrows and numbers indicate a possible message passing schedule.

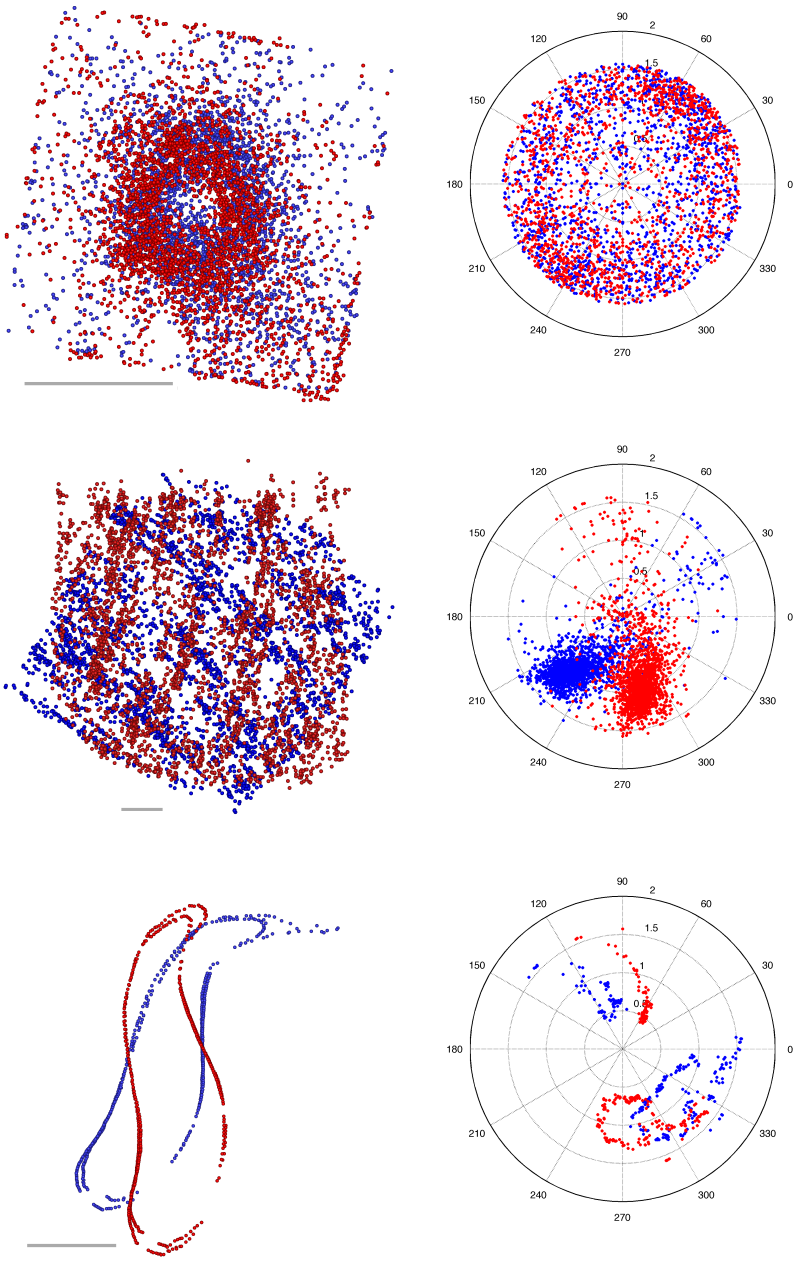

Figure 9. Endpoint positions and orientations of the evaluation samples. Endpoint positions and orientations on the facing boundaries of two consecutive unaligned sections. The different sections are indicated by color. Left: endpoint positions (view from top). Right: line orientations plotted in polar coordinates. Top: *C. elegans*. Middle: *X. laevis*. Bottom: *T. brucei*. Scale bars 2 μm.

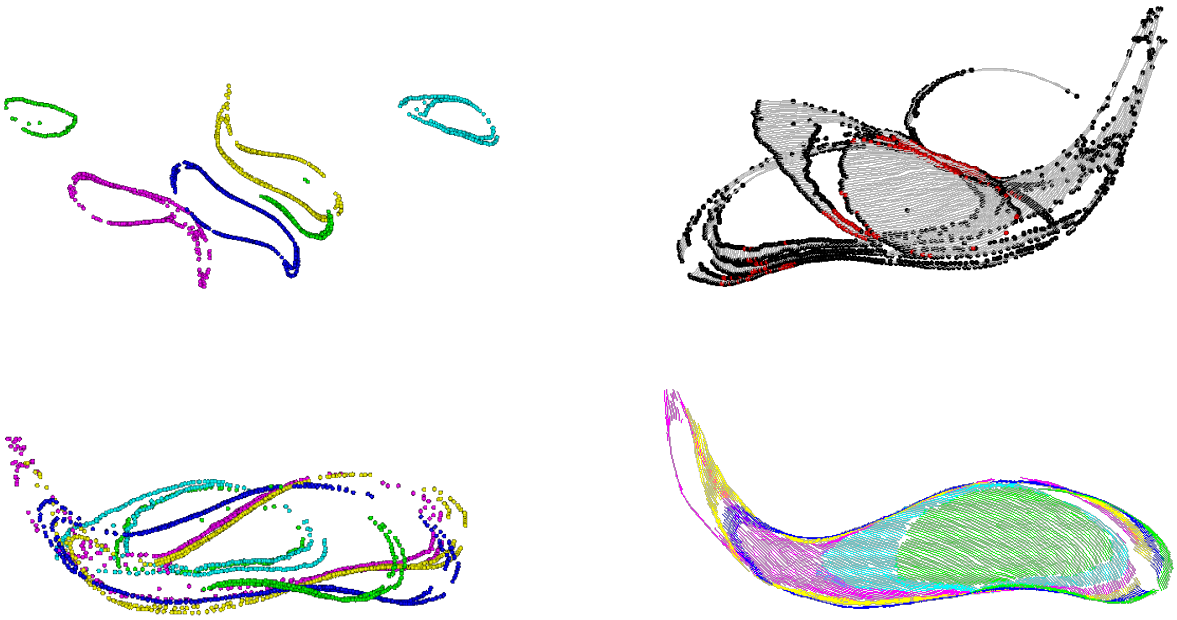

Figure 10. Alignment of *T. brucei*. The view is from the top. Unless stated otherwise, colors indicate 6 different sections. The stacking order is best seen in the final result (bottom right) at the right side: green, cyan, purple, yellow, blue, green. The length of the microtubule array is approximately 8 μm. Top left: initial position of unaligned sections. Only endpoints are displayed. Top right: The result after applying the initial alignment algorithm. Endpoints in all sections are depicted in black; lines in gray. The subset of endpoints that were used for computing the transformation are highlighted in red. Some pairs of sections are reasonably aligned. Other pairs are unaligned. Bottom left: endpoints after applying our algorithm for linear alignment from orientation (Figure S1). All sections are reasonably rotated. Bottom right: result after applying our algorithm for linear alignment from position and orientation (Figure S2). Endpoints and lines are displayed, both colored by section. All sections are reasonably aligned.

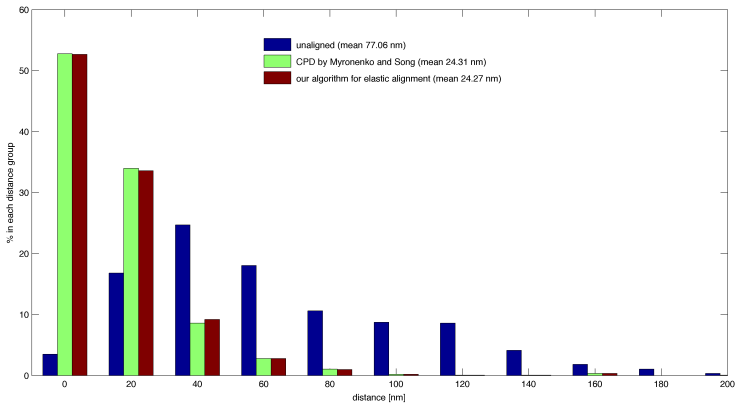

Figure 11. Distances before and after alignment. Histogram of the distances between corresponding endpoints for the *X. laevis* ground truth before and after applying the non-rigid point set registration algorithm by Myronenko and Song (Figure 4 in [28]) and our algorithm for elastic alignment.

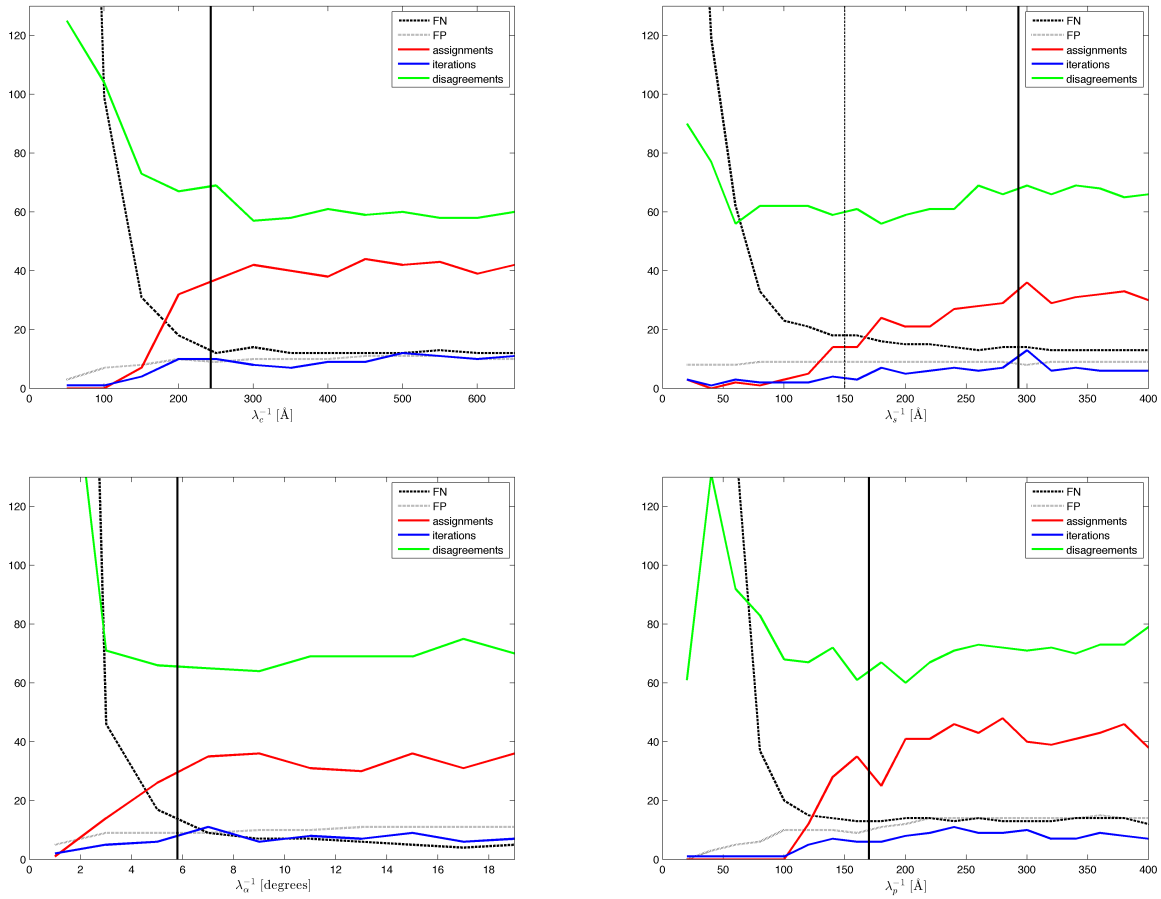

Figure 12. Matching performance over parameter variations. Displayed are false negatives (FN), false positives (FP), number of manual assignments, number of iterations, and number of disagreements when varying matching parameters around their maximum-likelihood estimates (indicated by solid vertical line) for the *X. laevis* sample: top left: by direct position distance parameter λ_c^{-1}; top right: by shift difference parameter λ_s^{-1} (the dashed vertical line indicates the value that we chose instead of the maximum-likelihood estimate); bottom left: by angle difference parameter λ_α^{-1}; bottom right: by projected distance parameter λ_p^{-1}.

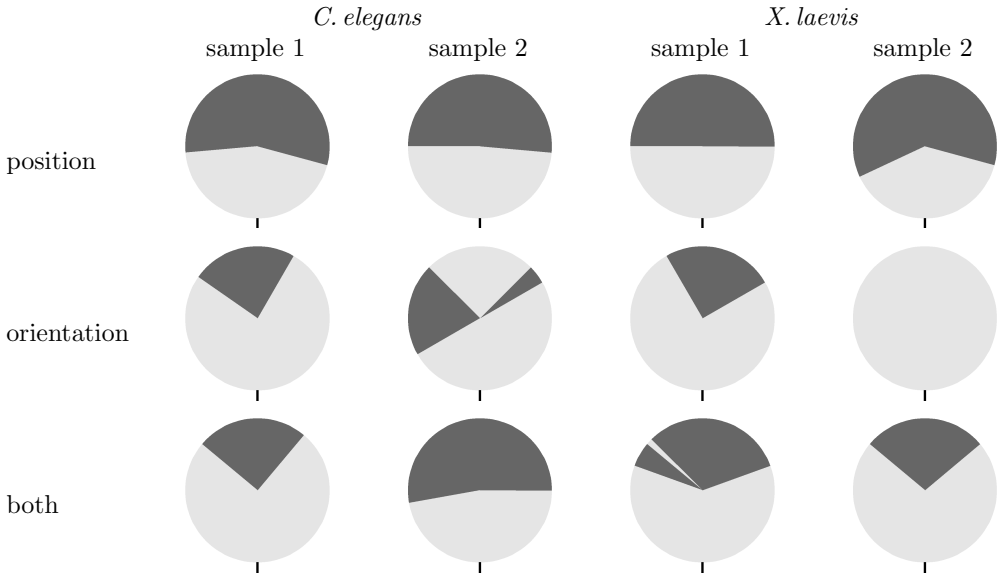

Figure 13. Stability of alignment. We rotated one section of a pair of correctly aligned sections for several samples over a range of 360° at 5° steps, as depicted by the circles, and tested whether the algorithms recovered the originally correct orientation, which is indicated by a vertical bar at the bottom of each circle. Dark gray indicates failure of the algorithm to recover the original rotation angle. Light gray indicates that the algorithm discovered the correct rotation. All algorithms found correct alignments for small rotations (light gray area in bottom half of each circle). The algorithm for linear alignment from orientation was most successful (largest light gray areas in middle row). Top: Rigid point set registration algorithm by Myronenko and Song (Figure 2 in [28]). Middle: Algorithm for linear alignment from orientation (Figure S1). Bottom: Algorithm for linear alignment from position and orientation (Figure S2).

Tables

Table 1. Comparison of probabilistic matching (PGM) and maximum weighted matching (MWM)

X. laevis

	PGM				MWM		
	P	R	DIS	Assign.	P	R	DIS
X1	0.956	0.951	0.038	4	0.901	0.904	0.088
X2	0.950	0.953	0.039	8	0.851	0.874	0.121
X3	0.956	0.948	0.038	19	0.609	0.634	0.349

C. elegans

	PGM				MWM		
	P	R	DIS	Assign.	P	R	DIS
	0.946	0.975	0.01	0	0.934	0.97	0.018

Precision P: ratio of correct pairs to total pairs found by the matching. Recall R: ratio of correct pairs to total pairs in the ground truth. Disagreement DIS: ratio of automatic matchings that disagree with the ground truth. Assign.: number of manual assignments. *X. laevis*: X1: after elastic alignment; PGM parameters $\lambda_c^{-1} = 243\,\text{Å}$, $\lambda_p^{-1} = 170\,\text{Å}$, $\lambda_\alpha^{-1} = 5.8\,°$, and $\lambda_s^{-1} = 150\,\text{Å}$. X2: after elastic alignment; PGM parameters $\lambda_c^{-1} = 486\,\text{Å}$, $\lambda_p^{-1} = 340\,\text{Å}$, $\lambda_\alpha^{-1} = 11.6\,°$, and $\lambda_s^{-1} = 150\,\text{Å}$. X3: after linear alignment, but before elastic alignment; PGM parameters $\lambda_c^{-1} = 771\,\text{Å}$, $\lambda_p^{-1} = 577\,\text{Å}$, $\lambda_\alpha^{-1} = 5.8\,°$, and $\lambda_s^{-1} = 150\,\text{Å}$. *C. elegans*: after elastic alignment; mean of three samples with ground truth by two experts; PGM parameters $\lambda_c^{-1} = 243\,\text{Å}$, $\lambda_p^{-1} = 170\,\text{Å}$, $\lambda_\alpha^{-1} = 5.8\,°$, and $\lambda_s^{-1} = 150\,\text{Å}$.

Supporting Information: Automated stitching of microtubule centerlines across serial electron tomograms

See main manuscript for references and figure legends.

Algorithms for fine alignment

This section describes in detail the algorithms for fine alignment. The method uses the expectation maximization algorithm. To apply expectation maximization, we derive 1) the update equations for the posterior $P(\mathbf{m}|\mathbf{x})$ in the E-step and 2) the update equations for the distribution and transformation parameters in the M-step, which is done by minimizing the expectation of the complete negative log-likelihood function L.

We give the detailed derivation of the update equations for each of the three methods and summarize the algorithms in Figures S1, S2 and S3 in the style of Myronenko and Song [28].

Notation In addition to the notation defined in the main text we will use the following notation: For two sets of endpoints $x_1, ...x_N$ and $y_1, ...y_M$ the matrix $\mathbf{X}$ contains all endpoint positions 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 $\mathbf{1}$.

Further prerequisites To account for outliers, Myronenko and Song introduce an additional term $P(\mathbf{x}, M+1) = 1/N$ to the joint distribution. This outlier term can be weighted with $0 \leq w \leq 1$. The marginal probability of $\mathbf{x}$ then takes the form:

$$P(\mathbf{x}) = \sum_{m=1}^{M+1} p(m)P(\mathbf{x}|m) = \frac{1-w}{M} \sum_{m=1}^{M} P(\mathbf{x}|m) + \frac{w}{N}. \tag{12}$$

We use the same outlier term in all three settings.

For convenience, we will often write sums as matrices using the following equalities:

$$\sum_{m,n=1}^{M,N} P(m|x_n)x_n = \mathbf{X}^\top \mathbf{P}^\top \mathbf{1} \tag{13}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n)y_m = \mathbf{Y}^\top \mathbf{P}\mathbf{1} \tag{14}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n)x_n^\top x_n = \mathrm{tr}(\mathbf{X}^\top \mathrm{diag}(\mathbf{P}^\top \mathbf{1})\mathbf{X}) \tag{15}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n)x_n x_n^\top = \mathbf{X}^\top \mathrm{diag}(\mathbf{P}^\top \mathbf{1})\mathbf{X} \tag{16}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) y_m^\top y_m = \mathrm{tr}(\mathbf{Y}^\top \mathrm{diag}(\mathbf{P1})\mathbf{Y}) \tag{17}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) y_m y_m^\top = \mathbf{Y}^\top \mathrm{diag}(\mathbf{P1})\mathbf{Y} \tag{18}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n^\top R y_m = \mathrm{tr}(\mathbf{X}^\top \mathbf{P}^\top \mathbf{Y} R^\top) \tag{19}$$

$$\sum_{m,n=1}^{M,N} P(m|x_n) x_n y_m^\top R^\top = \mathbf{X}^\top \mathbf{P}^\top \mathbf{Y} R^\top \tag{20}$$

Algorithm for linear alignment from orientation

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

$$P(\vec{\mathbf{m}}|\vec{\mathbf{x}}) = \frac{P(\vec{\mathbf{m}}, \vec{\mathbf{x}})}{P(\vec{\mathbf{x}})}. \tag{21}$$

To compute $P(\vec{\mathbf{m}}, \vec{\mathbf{x}})$, we add an outlier term $P(\vec{\mathbf{x}}|M+1) = 1/N$ and weight the priors with $0 \leq w \leq 1$ so that $P(\mathbf{m}) = (1-w)/M$ and $P(M+1) = w/N$. $P(\vec{\mathbf{m}}, \vec{\mathbf{x}})$ can be computed from the priors and $P(\vec{\mathbf{x}}|\vec{\mathbf{m}})$ which is given by Equation 6 for $m \leq M$. The joint distribution becomes:

$$P(\vec{\mathbf{m}}, \vec{\mathbf{x}}) = P(\vec{\mathbf{m}})P(\vec{\mathbf{x}}|\vec{\mathbf{m}}) = \frac{(1-w)}{M} \frac{\kappa}{2\pi(e^\kappa - e^{-\kappa})} \exp(\kappa(R\vec{y}_m)^\top \vec{\mathbf{x}}), \text{ for all } m \leq M \text{ and} \tag{22}$$

$$P(\vec{\mathbf{x}}, M+1) = w/N, \text{ for } m = M+1. \tag{23}$$

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

$$P(\vec{\mathbf{x}}) = P(\vec{\mathbf{x}}, M+1) + \sum_{m=1}^{M} P(\vec{\mathbf{m}}, \vec{\mathbf{x}}) = \frac{w}{N} + (1-w) \sum_{k=1}^{M} \frac{\kappa}{2M\pi(e^\kappa - e^{-\kappa})} \exp(\kappa(R\vec{y}_k)^\top \vec{\mathbf{x}}) \tag{24}$$

The posterior becomes

$$P(\vec{\mathbf{m}}|\vec{\mathbf{x}}) = \frac{P(\vec{\mathbf{m}}, \vec{\mathbf{x}})}{P(\vec{\mathbf{x}})} = \frac{\exp(\kappa \vec{x}_n^\top R\vec{y}_m)}{\sum_{k=1}^{M} \exp(\kappa \vec{x}_n^\top R\vec{y}_k) + \frac{wM}{(1-w)N} \frac{2\pi(\exp(\kappa) - \exp(-\kappa))}{\kappa}} \tag{25}$$

M-step In the M-step, we must write out the expectation of the negative log-likelihood function Q (Equation 3) and minimize it with respect to the transformation parameters R and the distribution parameter κ. Q becomes:

$$\begin{aligned} Q(R, \kappa) = -\kappa \sum_{m,n=1}^{M,N} p^{old}(m|\vec{x}_n) \vec{x}_n^\top R\vec{y}_m - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.} = \\ -\kappa \,\mathrm{tr}((\vec{\mathbf{X}}^\top \mathbf{P}^\top \vec{\mathbf{Y}})^\top R) - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.} \end{aligned} \tag{26}$$

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

$$\mathrm{tr}((\vec{\mathbf{X}}^\top \mathbf{P}^\top \vec{\mathbf{Y}})^\top R) = \mathrm{tr}((\vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d})^\top \tilde{R}) + \vec{\mathbf{X}}_z^\top \mathbf{P}^\top \vec{\mathbf{Y}}_z \tag{27}$$

where $\vec{\mathbf{X}}_{2d}$ is a matrix containing 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 $\vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d}$ (equivalently to Myronenko and Song [28], see Umeyama [42] and Myronenko and Song [43] for details on the method). κ cannot be obtained in closed form, but it can be computed with Newton's method. We iterate

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

$$\frac{\partial Q}{\partial \kappa} = -\operatorname{tr}((\vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d})^\top \tilde{R}) - \operatorname{tr}(\vec{\mathbf{X}}_z^\top \mathbf{P}^\top \vec{\mathbf{Y}}_z) - N_P(\frac{1}{\kappa} - \coth \kappa), \text{ and} \tag{28}$$

$$\frac{\partial^2 Q}{\partial^2 \kappa} = -N_P(-\frac{1}{\kappa^2} + \frac{1}{\sinh \kappa^2}). \tag{29}$$

Because we only compute a rotation, the endpoints are not aligned. To align them, we first note that even though we did not explicitly take the positions into account, the likelihood that two orientations match also applies to the corresponding endpoint positions. To compute an alignment of the positions, we compute the squared distances of the positions, 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 Figure S1.

Algorithm for linear alignment from position and orientation

E-step The joint distribution of $\mathbf{x}, \vec{\mathbf{x}}$ and $\mathbf{m}, \vec{\mathbf{m}}$ is (Equation 2 and 6):

$$P(\mathbf{x}, \vec{\mathbf{x}}, \mathbf{m}, \vec{\mathbf{m}}) = \mathcal{N}(\mathbf{x}; \mathbf{m}, \sigma^2)\mathcal{F}_3(\vec{\mathbf{x}}; \vec{\mathbf{m}}, \kappa)\frac{1}{M}\delta_{\text{ind}(\mathbf{m}),\text{ind}(\vec{\mathbf{m}})} \tag{30}$$

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 as in the previous algorithm and weight the priors as before with $0 \leq w \leq 1$. The posterior then is:

$$\begin{aligned} P(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}}) &= \frac{P(\mathbf{x}, \vec{\mathbf{x}}, \mathbf{m}, \vec{\mathbf{m}})}{P(\mathbf{x}, \vec{\mathbf{x}})} = \\ &\frac{(1-w)\mathcal{N}(\mathbf{x}; \mathbf{m}, \sigma^2)\mathcal{F}(\vec{\mathbf{x}}; \vec{\mathbf{m}}, \kappa)\frac{1}{M}\delta_{\text{ind}(\mathbf{m}),\text{ind}(\vec{\mathbf{m}})}}{\frac{w}{N} + \sum_{k=1}^{M}\mathcal{N}(\mathbf{x}; y_k, \sigma^2)\mathcal{F}(\vec{\mathbf{x}}; \vec{y}_k, \kappa)\frac{1}{M}\delta_{k,k}} = \\ &\frac{\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_m - t||^2)\exp(\kappa \vec{x}_n^\top R\vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_k - t||^2)\exp(\kappa \vec{x}_n^\top R\vec{y}_k)) + \frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}} \end{aligned} \tag{31}$$

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

$$\begin{aligned} Q(s,\tilde{R},t,\sigma^2,\kappa) =& \frac{1}{2\sigma^2}\sum_{m,n=1}^{M,N} p^{old}(y,\vec{y}|x,\vec{x})||x_n - s\tilde{R}y_m - t||^2 + N_P \log\sigma \\ & -\kappa \sum_{m,n=1}^{M,N} p^{old}(y,\vec{y}|x,\vec{x})\vec{x}_n^\top R\vec{y}_m - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.} \end{aligned} \tag{32}$$

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 [28])

$$t = \frac{1}{N_P}\sum_{m,n=1}^{M,N} p^{old}(y_m,\vec{y}_m|x_n,\vec{x}_n)x_n - \frac{1}{N_P}\sum_{m,n=1}^{M,N} p^{old}(y_m,\vec{y}_m|x_n,\vec{x}_n)\tilde{R}y_n = \mu_x - s\tilde{R}\mu_y \tag{33}$$

We then plug t back into Equation 32. The log-likelihood becomes

$$\begin{aligned} Q(s,\tilde{R},\sigma^2,\kappa) =& -\frac{1}{2\sigma^2}\sum_{m,n=1}^{M,N} p^{old}(y,\vec{y}|x,\vec{x})||\hat{x}_n - s\tilde{R}\hat{y}_m||^2 + \frac{N_P}{2}\log\sigma \\ & -\kappa \sum_{m,n=1}^{M,N} p^{old}(y,\vec{y}|x,\vec{x})\vec{x}_n^\top R\vec{y}_m - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.}, \end{aligned} \tag{34}$$

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:

$$\begin{aligned} Q(s,\tilde{R},\sigma^2,\kappa) =& +\frac{1}{2\sigma^2}(\operatorname{tr}(\hat{\mathbf{X}}^\top \operatorname{diag}(\mathbf{P}^\top\mathbf{1})\hat{\mathbf{X}}) - 2s\cdot\operatorname{tr}((\hat{\mathbf{X}}^\top\mathbf{P}^\top\hat{\mathbf{Y}})^\top\tilde{R}) + s^2\operatorname{tr}(\hat{\mathbf{Y}}^\top\operatorname{diag}(\mathbf{P1})\hat{\mathbf{Y}})) - \\ & -\kappa\operatorname{tr}((\vec{\mathbf{X}}^\top\mathbf{P}^\top\vec{\mathbf{Y}})^\top R) + N_P\log\sigma^2 - N_P\log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.} \end{aligned} \tag{35}$$

We could not find a closed form solution for any of the parameters. Instead we search the optimum for the parameters numerically, except for t, which we already eliminated. In general, using numerical optimization in this case is problematic due to the constraints $\tilde{R}^\top\tilde{R} = \mathbf{I}$ 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 35, but it is tedious work. For the convenience of the reader, we give details on the derivative with respect to ρ below (Section Q derivatives) and list all first and second order derivatives in Figure S14. Figure S2 summarizes the algorithm.

The numerical optimization in the M-step 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 [44]. 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 use other approaches that use scaled vectors such as Wells [26], because our orientations are unit vectors.

Algorithm for elastic alignment

For the elastic transformation model, y is transformed via a displacement vector $\nu(y)$ (Myronenko and Song [28])

$$T(Y,\nu) = Y + \nu(Y). \tag{36}$$

Here, the task is to find a function ν that gives enough flexibility to capture the deformation of the data but still maintains the overall neighborhood structure of the points.

Myronenko and Song derive the displacement vectors by solving the regularized expectation of the negative log-likelihood function

$$f(\nu, \sigma^2) = Q_c(\nu, \sigma^2) + \frac{\lambda}{2}\Phi(\nu), \tag{37}$$

where Φ penalizes high frequencies of ν. See [28] for a detailed derivation of the update equations for the M-step. The resulting vectors $\nu(y)$ are the product of an $M \times M$ smoothing matrix $\mathbf{G}$ and, for the 2D case, 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 and stays fixed.

M-step Coupling the transformation of positions and orientations is not obvious, because the transformation is formulated as position displacement vectors ν (Equation 36) and the rotational part of the transformation is not explicitly accessible. In practice, we assume that the elastic transformation contains only a minimal rotation that can be ignored, because we have applied a linear alignment before. With this assumption we can rewrite Equation 37 with the orientation as

$$f(\nu, \sigma^2, \kappa) = Q_c(\nu, \sigma^2) + \frac{\lambda}{2}\Phi(\nu) + Q_d(\kappa), \text{ with} \tag{38}$$

$$Q_d(\kappa) = -\kappa \sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m | x_n, \vec{x}_n)\vec{x}_n^\top \vec{y}_m - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) + \text{const.}, \text{ and} \tag{39}$$

$$Q_c(\sigma^2) = \frac{1}{2\sigma^2} \sum_{m,n=1}^{M,N} p^{old}(y_m, \vec{y}_m | x_n, \vec{x}_n)||x_n - (y_m - \nu(y_m))||^2 + N_P \log \sigma^2 + \text{const.} \tag{40}$$

Because Q_d is independent of the displacement vectors μ, the optimization can be performed exactly as described in Myronenko and Song [28]. 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. 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

$$P(\mathbf{m}, \vec{\mathbf{m}} | \mathbf{x}, \vec{\mathbf{x}}) = \frac{P(\mathbf{x}, \vec{\mathbf{x}}, \mathbf{m}, \vec{\mathbf{m}})}{P(\mathbf{x}, \vec{\mathbf{x}})} = \frac{\exp(-\frac{1}{2\sigma^2}||x - (y_m + (\mathbf{G}(m, \cdot)\mathbf{W})^\top)||^2) \exp(\kappa \vec{x}^\top \vec{y}_m)}{\sum_{k=1}^{M} \exp(-\frac{1}{2\sigma^2}||x - (y_k + (\mathbf{G}(m, \cdot)\mathbf{W})^\top)||^2) \exp(\kappa \vec{x}^\top \vec{y}_k)) + \frac{wM}{(1-w)N} \frac{4\pi^2\sigma^2(\exp(\kappa) - \exp(-\kappa))}{\kappa}} \tag{41}$$

The algorithm is summarized in Figure S3.

Implementation details

To optimize Q for the linear alignment from position and orientation (Algorithm S2), we use IPOpt [48], a free optimization library. As the 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 (see Figure S13). To avoid this, we initialize σ^2 as

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

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 that is the minimum of the objective function becomes (see Figure S13). 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 does not always find an optimum in later expectation maximization iterations in our implementation. However, for our test data, we found that about 4000 iterations in IPOpt were sufficient to find a new parameter set that decreased the log-likelihood.

The running time of the algorithm for elastic alignment is $O(M^3)$. It can can reduced to linear by computing P with a fast Gauss transform and G by a low rank approximation as described by Myronenko and Song [28]. We did not implement this technique, which is why our implementation has cubic running time.

Q derivatives

Splitting the negative log-likelihood function (Equation 35) again into terms dependent and independent of $\tilde{R}$ (by Equation 4), Q can be written as :

$$\begin{aligned} Q(s, \tilde{R}, \sigma^2, \kappa) = \frac{1}{2\sigma^2}(&\operatorname{tr}(\hat{\mathbf{X}}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\hat{\mathbf{X}}) - 2s \cdot \operatorname{tr}((\hat{\mathbf{X}}^\top \mathbf{P}^\top \hat{\mathbf{Y}})^\top \tilde{R}) + s^2 \operatorname{tr}(\hat{\mathbf{Y}}^\top \operatorname{diag}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}}))- \\ &\kappa(\operatorname{tr}((\vec{\mathbf{X}}_{2d}^{\top} \mathbf{P}^\top \vec{\mathbf{Y}}_{2d})^\top \tilde{R}) + \vec{\mathbf{X}}_z^{\top} \mathbf{P}^\top \vec{\mathbf{Y}}_z) \\ &N_P \log \sigma^2 - N_P \log(\frac{\kappa}{e^\kappa - e^{-\kappa}}) \end{aligned} \tag{42}$$

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

$$\begin{aligned} Q(s, \tilde{R}, \sigma^2, \kappa) = \frac{1}{2\sigma^2}(&\operatorname{tr}(A) - 2s \cdot \operatorname{tr}(B^\top \tilde{R}) + s^2 \operatorname{tr}(C))- \\ &\kappa(\operatorname{tr}(D^\top \tilde{R}) + \operatorname{tr}(E))+ \\ &N_P \log \sigma^2 - N_P \log(\frac{\kappa}{2\pi(e^\kappa - e^{-\kappa})}) + \text{const.} \end{aligned} \tag{43}$$

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 with respect to this angle we find that

$$\frac{\partial \operatorname{tr}(K^\top \tilde{R})}{\partial \rho} = \operatorname{tr}(K^\top \frac{\partial \tilde{R}}{\partial \rho}) \tag{44}$$

and

$$\frac{\partial \tilde{R}}{\partial \rho} = \tilde{R}' = \begin{pmatrix} -\sin\rho & -\cos\rho \\ \cos\rho & -\sin\rho \end{pmatrix} \tag{45}$$

The second derivative can be found equivalently.

$$\frac{\partial^2 \tilde{R}}{\partial^2 \rho^2} = \tilde{R}'' = \begin{pmatrix} -\cos\rho & \sin\rho \\ -\sin\rho & -\cos\rho \end{pmatrix} \tag{46}$$

We give the partial derivatives of Equation 43 with respect to s, ρ, σ^2 and κ in Figure S14.

Supporting Figures

Algorithm for linear alignment from orientation

Initialization: $R = \mathbf{I}, 0 \leq w \leq 1, 0 < \kappa \ll 1$

Repeat until convergence:

- E-step: Compute posterior matrix $\mathbf{P}_{nm} = \frac{\exp(\kappa \vec{x}_n^\top R \vec{y}_m)}{\sum_{k=1}^{M} \exp(\kappa \vec{x}_n^\top R \vec{y}_k)) + \frac{wM}{(1-w)N} \frac{2\pi(\exp(\kappa) - \exp(-\kappa))}{\kappa}}$
- 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 positions: $\mu_x = \frac{1}{N_p} \mathbf{X}^\top \mathbf{P}^\top \mathbf{1}, \mu_y = \frac{1}{N_p} \mathbf{Y}^\top \mathbf{P} \mathbf{1}, t = \mu_x - \tilde{R} \mu_y, \mathbf{Y}^{final} = \mathbf{Y} \tilde{R}^\top + \mathbf{1} t^\top$

Figure S1. Algorithm for linear alignment from orientation.

Algorithm for linear alignment from position and orientation

Initialization: $R = \mathbf{I}, s = 1, t = \mathbf{0}, 0 \leq w \leq 1, \sigma^2 = \frac{1}{2NM}\sum_{m,n=1}^{M,N} ||x_n - y_m||^2, 0 < \kappa \ll 1$

Repeat until convergence:

- E-step: Compute $\mathbf{P}_{nm} = \frac{\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_m - t||^2)\exp(\kappa \vec{x}_n^\top R \vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n - s\tilde{R}y_k - t||^2)\exp(\kappa \vec{x}_n^\top R \vec{y}_k)) + \frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}}$
- M-step: Solve for $\tilde{R}, s, t, \sigma^2$ and κ
  - $N_P = \mathbf{1}^\top \mathbf{P} \mathbf{1}$
  - $\mu_x = \frac{1}{N_p}\mathbf{X}^\top \mathbf{P}^\top \mathbf{1}$, $\mu_y = \frac{1}{N_p}\mathbf{Y}^\top \mathbf{P}^\top \mathbf{1}$
  - $\hat{\mathbf{X}} = \mathbf{X} - \mathbf{1}\mu_x^\top, \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{1}\mu_y^\top$
  - Numerically minimize $Q(s, \tilde{R}, \sigma^2, \kappa) =$
    $\frac{1}{2\sigma^2}(\mathrm{tr}(\hat{\mathbf{X}}^\top \mathrm{diag}(\mathbf{P}^\top \mathbf{1})\hat{\mathbf{X}}) - 2s \cdot \mathrm{tr}((\hat{\mathbf{X}}^\top \mathbf{P}^\top \hat{\mathbf{Y}})^\top \tilde{R}) + s^2 \mathrm{tr}(\hat{\mathbf{Y}}^\top \mathrm{diag}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}})) -$
    $\kappa(\mathrm{tr}((\vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d})^\top \tilde{R}) + \vec{\mathbf{X}}_z^\top \mathbf{P}^\top \vec{\mathbf{Y}}_z) + N_P \log \sigma^2 - N_P \log(\frac{\kappa}{2\pi(e^\kappa - e^{-\kappa})})$
  - $t = \mu_x - s\tilde{R}\mu_y$

Compute orientations: $\vec{\mathbf{Y}}^{final} = \vec{\mathbf{Y}}R^\top$

Compute positions: $\mathbf{Y}^{final} = s\mathbf{Y}\tilde{R}^\top + \mathbf{1}t^\top$

Figure S2. Algorithm for linear alignment from position and orientation.

Algorithm for elastic alignment

Initialization: $\mathbf{W} = 0, \sigma^2 = \frac{1}{2NM}\sum_{m,n=1}^{M,N} ||x_n - y_n||^2, 0 < \kappa \ll 1$

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 $\mathbf{P}_{mn} = \frac{\exp(-\frac{1}{2\sigma^2}||x_n - (y_m + (\mathbf{G}(m,\cdot)\mathbf{W})^\top)||^2)\exp(\kappa \vec{x}_n^\top \vec{y}_m)}{\sum_{k=1}^{M}\exp(-\frac{1}{2\sigma^2}||x_n - (y_k + (\mathbf{G}(m,\cdot)\mathbf{W})^\top)||^2)\exp(\kappa \vec{x}_n^\top \vec{y}_k)) + \frac{wM}{(1-w)N}\frac{4\pi^2\sigma^2(\exp(\kappa)-\exp(-\kappa))}{\kappa}}$
- M-step:
  - Solve $(\mathbf{G} + \lambda\sigma^2 \mathrm{diag}(\mathbf{P}\mathbf{1})^{-1})\mathbf{W} = \mathrm{diag}(\mathbf{P}\mathbf{1})^{-1}\mathbf{P}\mathbf{X} - \mathbf{Y}$ for $\mathbf{W}$
  - $N_P = \mathbf{1}\mathbf{P}^\top \mathbf{1}$, $\mathbf{T} = \mathbf{Y} + \mathbf{G}\mathbf{W}$
  - $\sigma^2 = \frac{1}{2N_P}(\mathrm{tr}(\mathbf{X}^\top \mathrm{diag}(\mathbf{P}^\top \mathbf{1})\mathbf{X}) - 2\,\mathrm{tr}((\mathbf{P}\mathbf{X})^\top \mathbf{T}) + \mathrm{tr}(\mathbf{T}^\top \mathrm{diag}(\mathbf{P}\mathbf{1})\mathbf{T}))$
  - iterate $\kappa = \kappa - \frac{-\mathrm{tr}((\vec{\mathbf{X}}^\top \mathbf{P}^\top \vec{\mathbf{Y}})^\top) - N_P(\frac{1}{\kappa} - \coth\kappa)}{-N_P(-\frac{1}{\kappa^2} + \frac{1}{\sinh^2\kappa})}$, until convergence

Compute positions: $\mathbf{Y}^{final} = \mathbf{Y} + \mathbf{G}\mathbf{W}$

Figure S3. Algorithm for elastic alignment. The differences to the non-rigid point set registration algorithm by Myronenko and Song [28] (Figure 4 therein for two dimension $D = 2$) are the computation of the posterior $p^{old}(\mathbf{m}, \vec{\mathbf{m}}|\mathbf{x}, \vec{\mathbf{x}})$ and the update of κ.

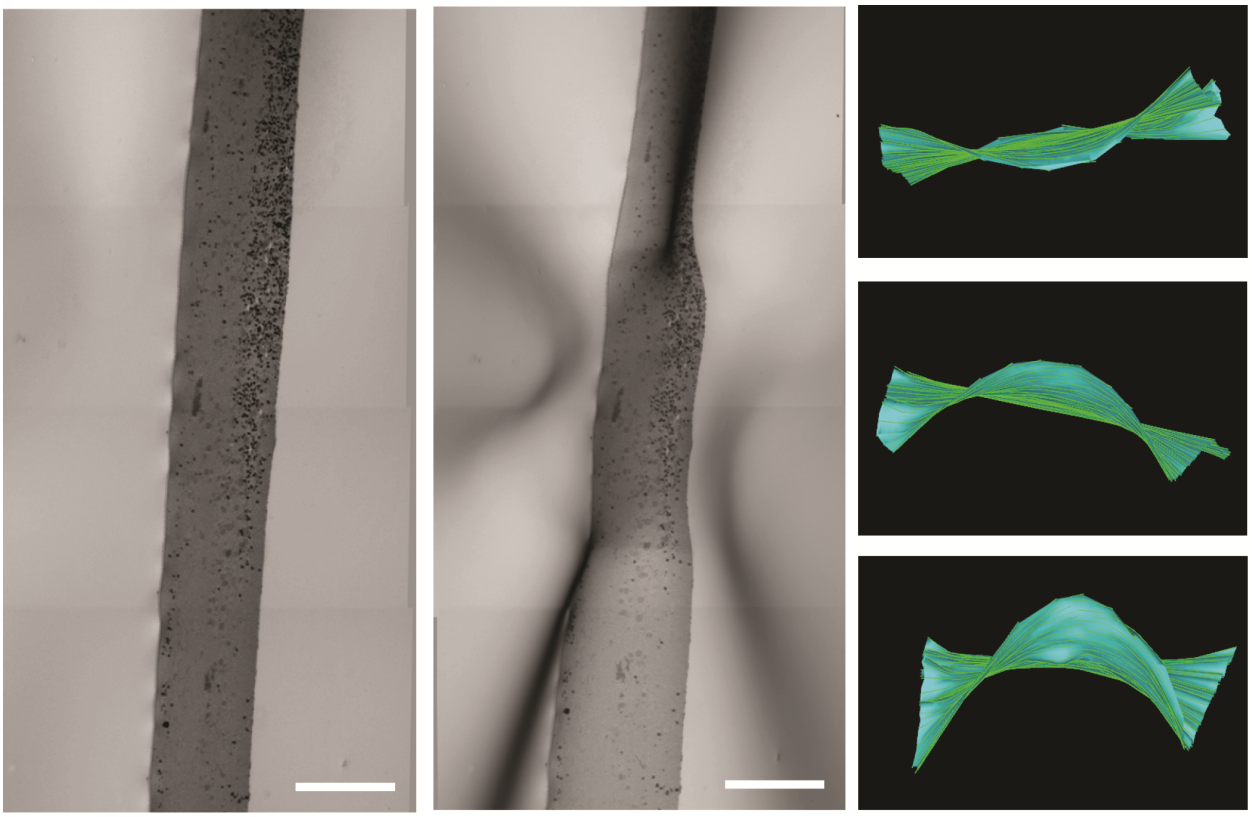

Figure S4. Deformations caused by electron beam exposure. *X. laevis* sample; scale bars 10 μm. Left: Low-resolution image before acquisition. Middle: Same sample after an exposure time of 11 hours, which is necessary for multiple area acquisition. The sample is substantially damaged. Right: Surface that represents boundary of relevant image signal in the reconstructed volume. The surface was manually outlined using IMOD [13]. The tomogram for the surface in the topmost panel was acquired first and consequently has the least amount of deformation. The tomogram for the surface in the bottommost panel was acquired last and consequently has the largest amount of deformation.

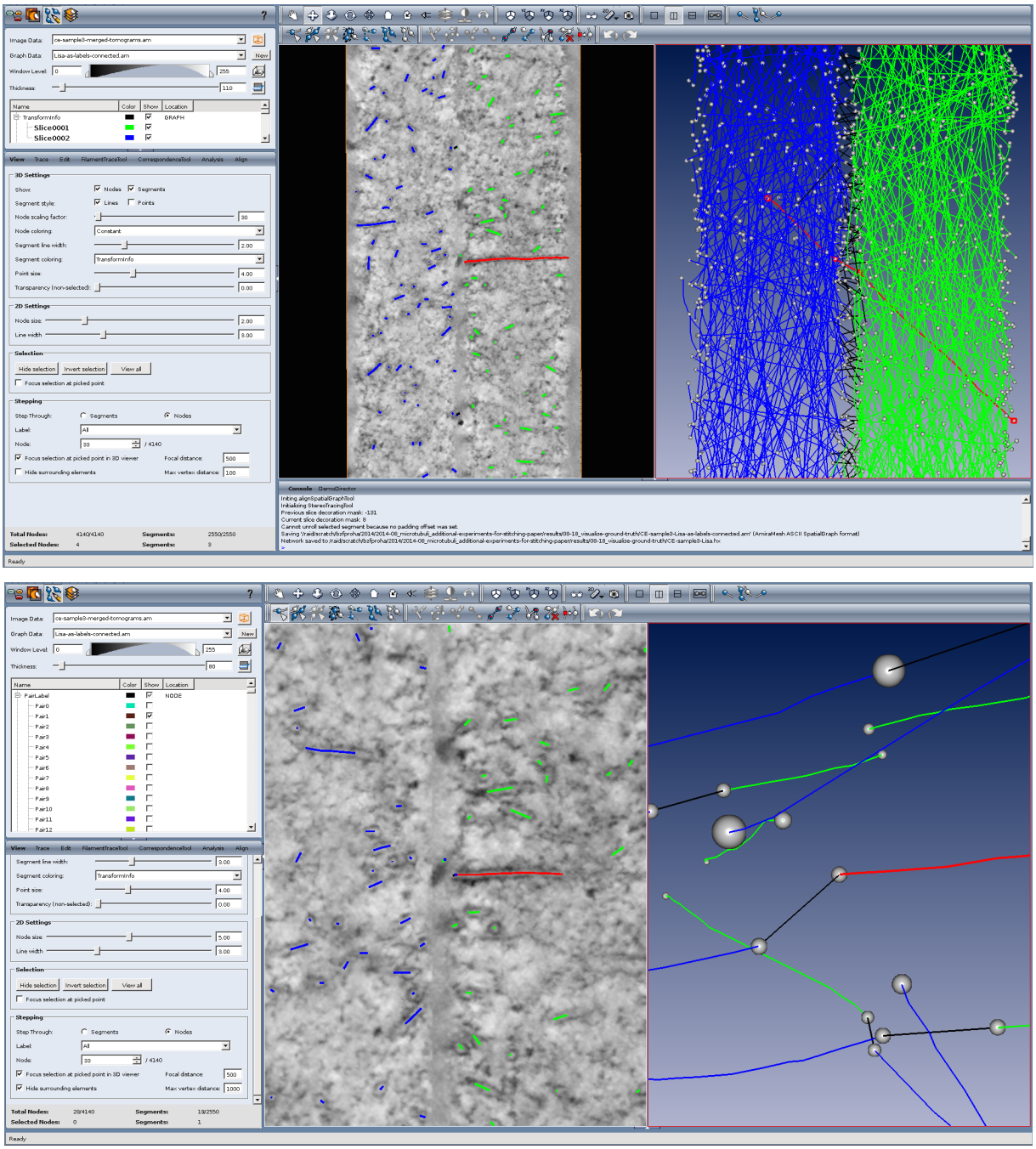

Figure S5. Graphical user interface for manual inspection and correction. The main interface elements are a perspective view of the line geometry (right) and a view that shows an oblique image slice together with surrounding lines (center). Lines from two different sections are indicated in green and blue. Black lines indicate connections across the section boundary. Endpoints are displayed in grey. The user interface elements on the left control line display parameters and support navigation, such as adjusting the views to a specific endpoint. Top: The perspective view is adjusted to show the full thickness of both sections. A line and its continuation in the neighboring section are highlighted in red. The line ends in the middle of the blue section, which probably indicates a natural microtubule end. Bottom: Closeup view as it is typically used for inspection. The user interface automatically adjusts the view to an endpoint and the surrounding lines. The operator can then inspect and make modifications. The same line as in the top panel is highlighted in red, but here only in one section.

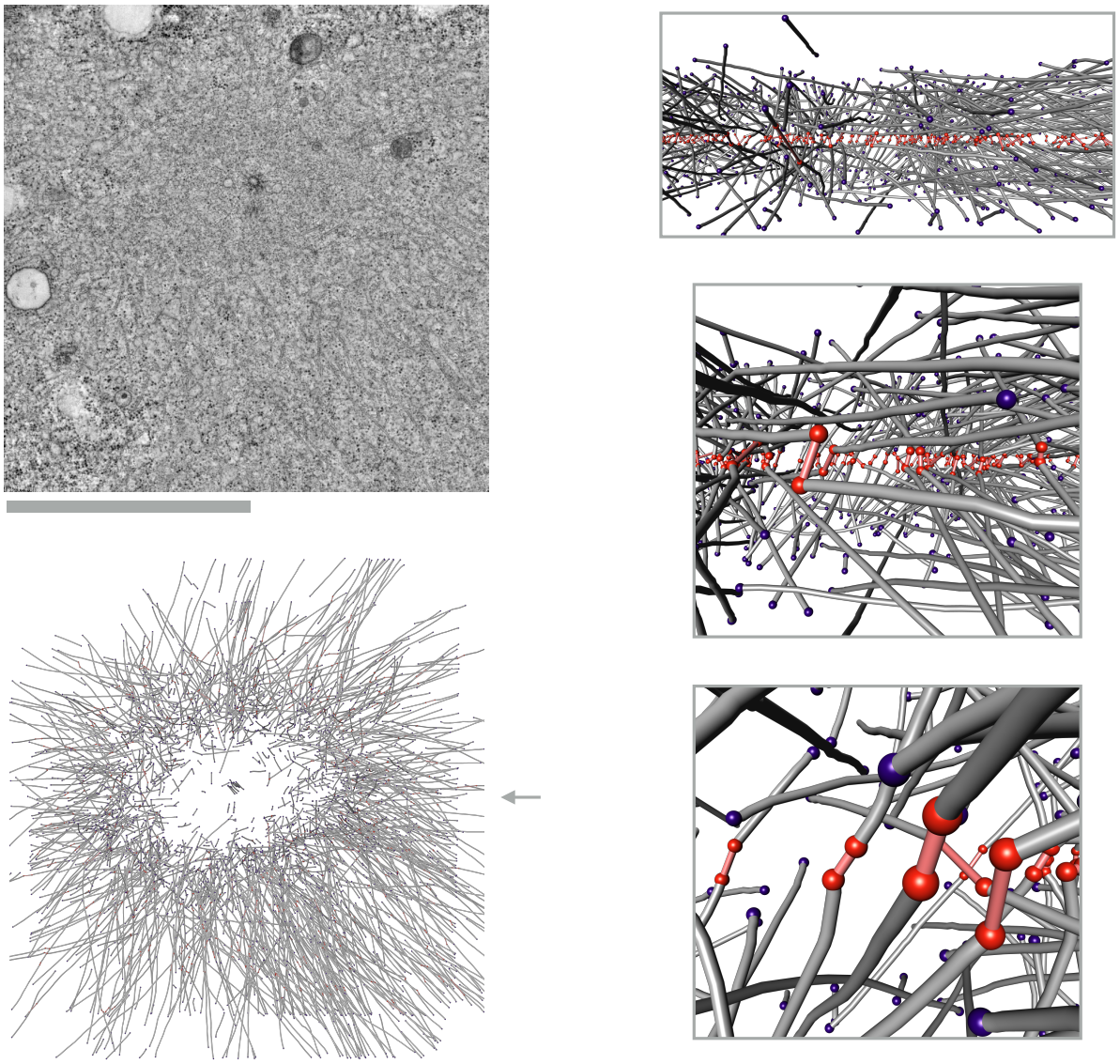

Figure S6. *C. elegans* tomogram and ground truth for evaluation. Top left: Slice through tomogram (scale bar 2 μm). Bottom left: View from top on ground truth microtubule centerlines for two consecutive sections. Right: View from the right side (as indicated by the gray arrow) at the ground truth microtubule centerlines with increasingly closer views from top to bottom. The depth of view has been restricted by a clipping plane to reduce overdrawing. Connections across the section boundary are indicated in red (endpoints and connecting lines). Blue endpoints are unconnected; microtubules probably naturally end there within a section.

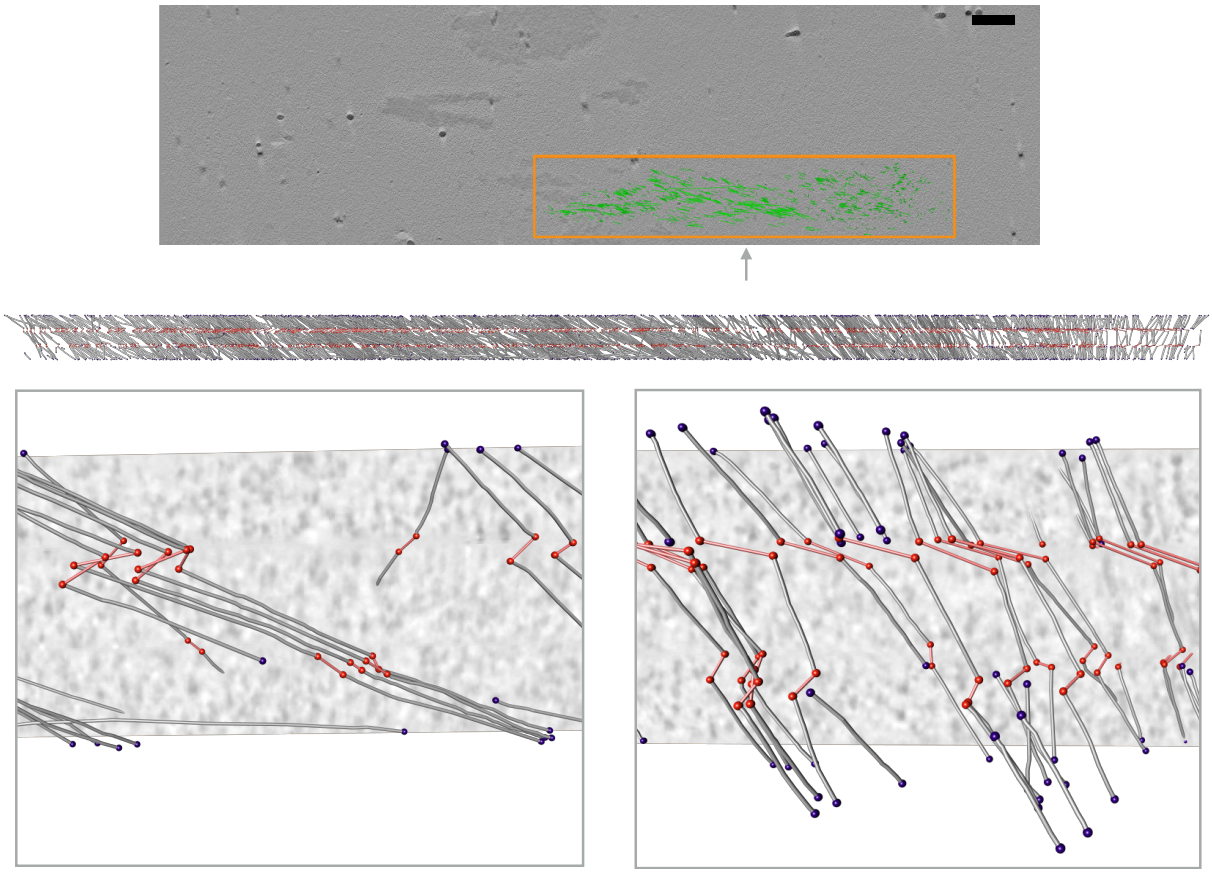

Figure S7. *X. laevis* tomogram and ground truth for evaluation. Top: Slice through a tomogram (scale bar 2 μm). The orange box indicates the region for which a ground truth was prepared. Middle: View from side (as indicated by the gray arrow) at the ground truth microtubule centerlines for three consecutive sections inside the orange box. Bottom: Two closeup views. The depth of view has been restricted by a vertical slice through the tomogram to avoid overdrawing (visible in light gray in the background; some lines are partially hidden). Connections across section boundaries are indicated in red (endpoints and connecting lines). Blue endpoints are unconnected; microtubules probably naturally end there within a section, or the corresponding microtubule centerline in the next section is missing.

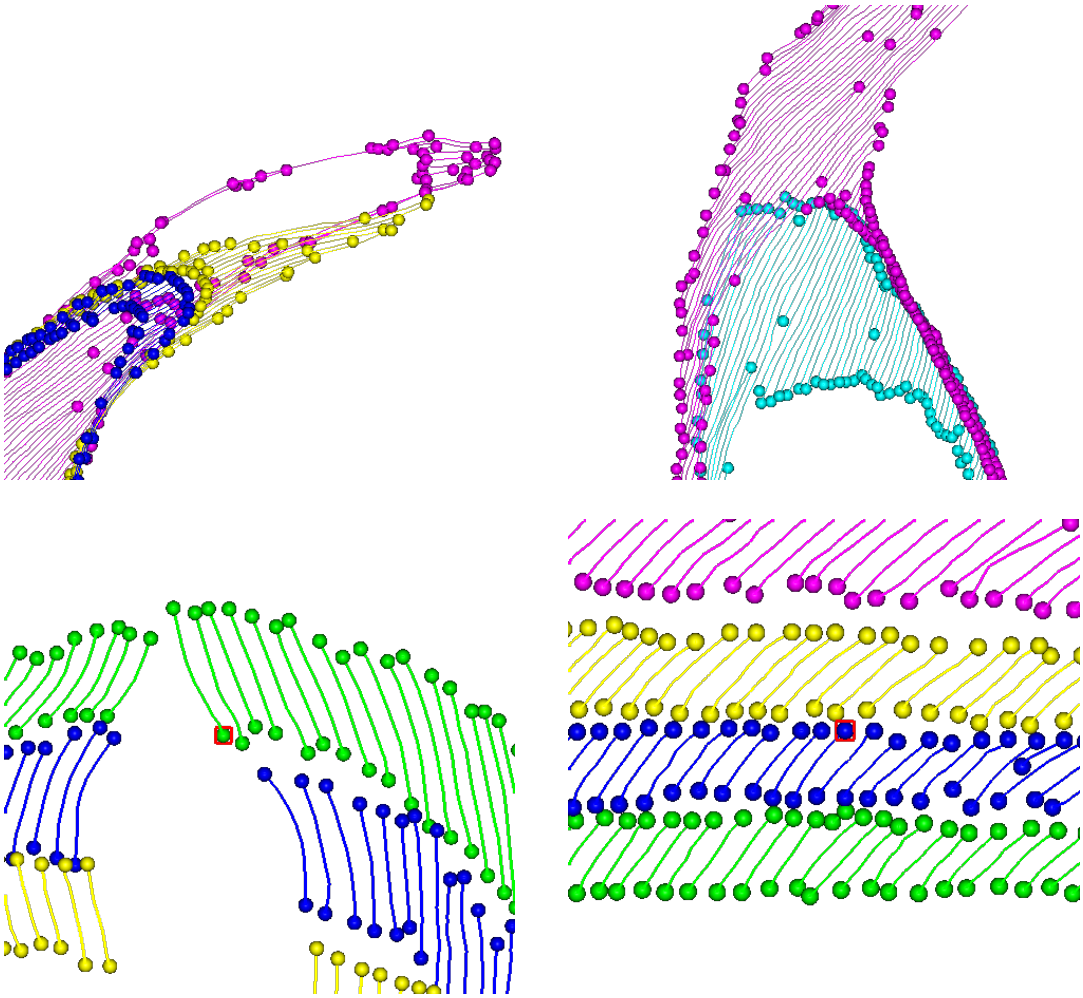

Figure S8. Alignment and matching of *T. brucei* samples. Colors indicated different sections. Top: View from top onto results after applying the algorithm for elastic alignment (Figure S3). Some endpoints are obviously not properly aligned (yellow endpoints on left; cyan on right). The reason is probably that endpoints do not outline the same shape in each section, because the outline varies substantially from section to section. Bottom: View from side at sections for which user input during matching was requested. Red boxes indicate two critical nodes that an expert would have to assign manually. The correct continuation between sections is hard to impossible to decide locally. Bottom left: The continuation is obviously unclear. Bottom right: Assuming the small blue line that is visible close to the right image border is assigned to the yellow endpoint right above, then the assignment at the red box in the center is unclear.

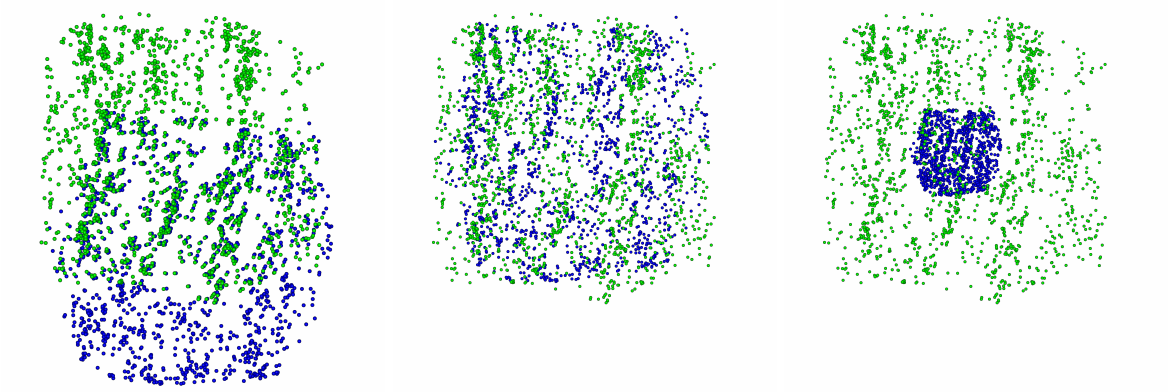

Figure S9. Alignment failure on sections with partial overlap. Left: example of two sections with partial overlap. Middle: Result of applying the rigid point set registration algorithm by Myronenko and Song (Figure 2 in [28]). Right: Result of applying our algorithm for linear alignment from position and orientation (Figure S2). Both algorithms failed to compute a reasonable alignment.

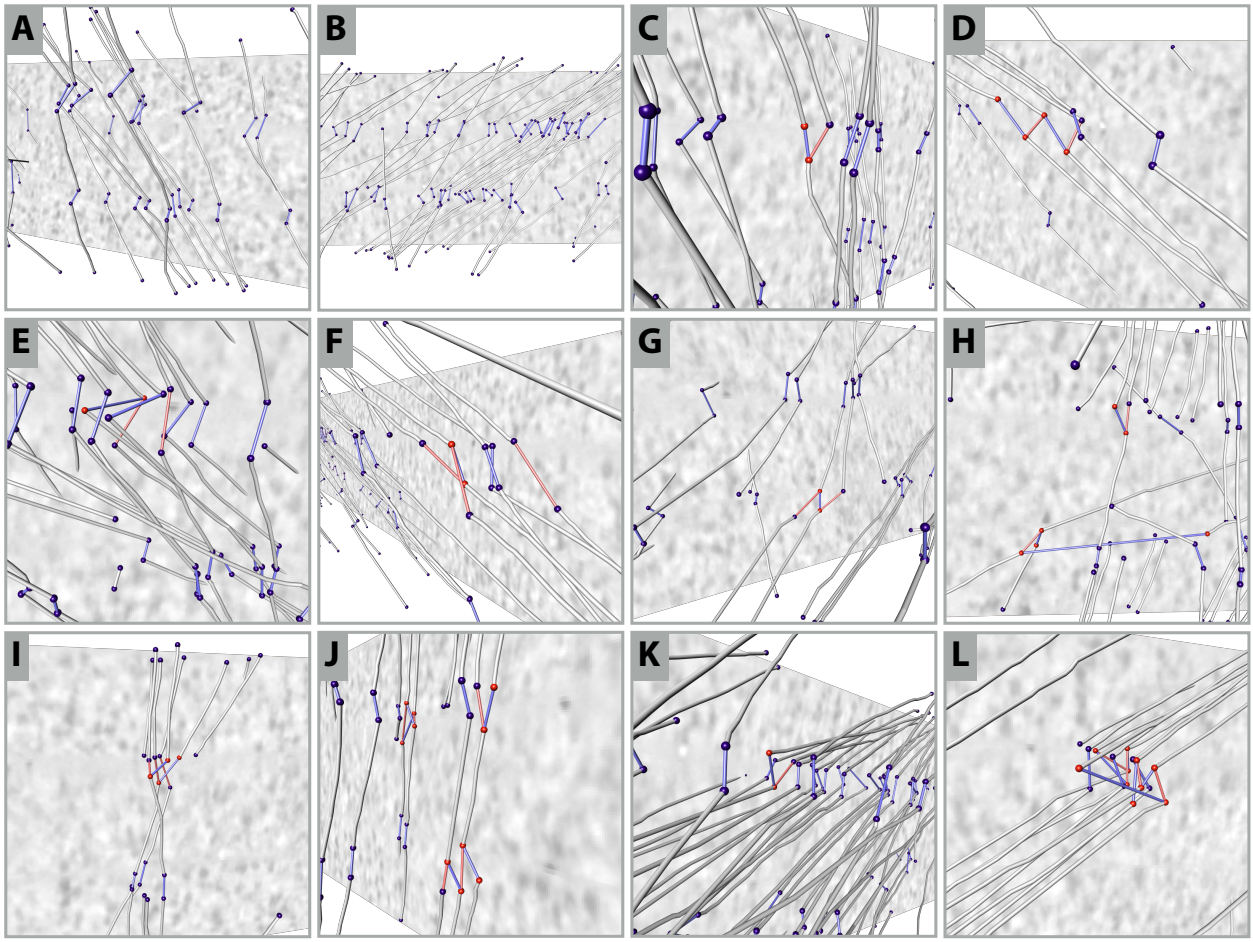

Figure S10. Comparison with ground truth for *X. laevis* sample. Views at the ground truth in comparison to the PGM matching that was computed after the algorithm for elastic alignment had been applied. In-section centerlines are grey. Endpoints that agree between the ground truth and the PGM matching are indicated in blue. Endpoints that are connected in the ground truth and connected differently in the PGM matching are indicated in red. Ground truth connections are indicated in blue. Connections that are in the PGM matching but not in the ground truth are indicated in red. A red connection with two blue endpoints indicates that the connection is in the PGM matching but not in the ground truth (false positive). The depth of view has been restricted by a vertical slice through the tomogram to avoid overdrawing (visible in light gray in the background; some lines are partially hidden). A, B: all connections agree. C, D: the PGM chose different connections. E, F, G: the PGM made additional connections. H: rare situation with a line running nearly horizontally that has a long connection in the ground truth but a short connection in the PGM matching. I, J: the top and bottom section contain more lines than the middle section; lines are probably missing in the middle section. K, L: more complex situation with bundles of many parallel lines.

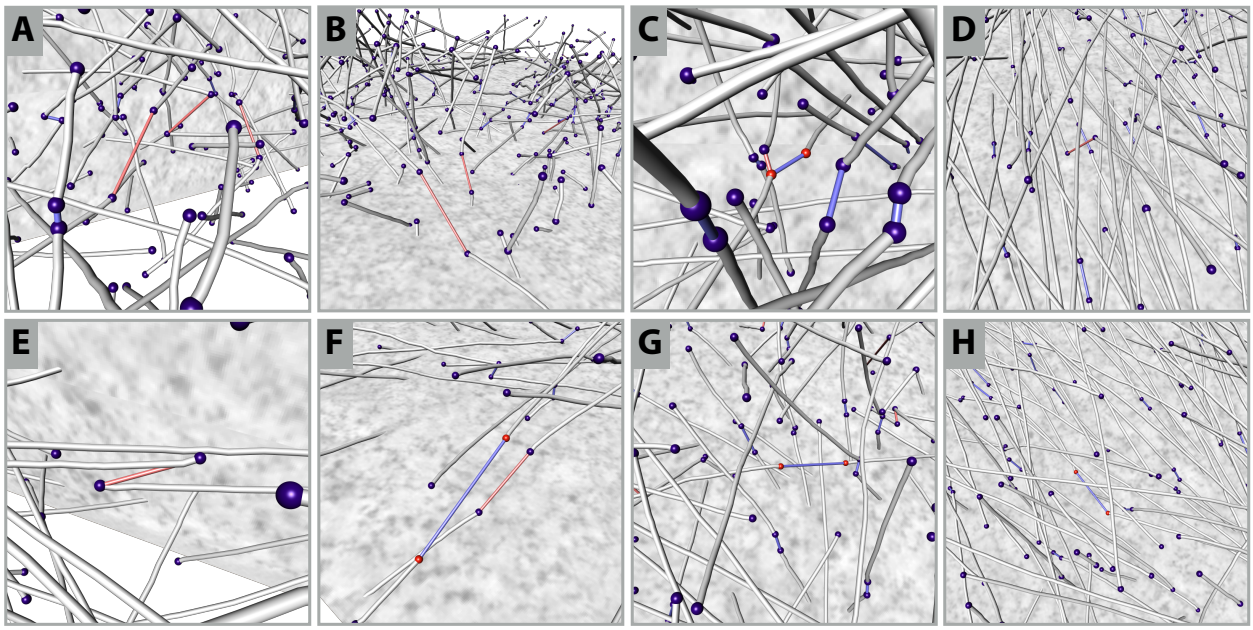

Figure S11. Comparison with ground truth for *C. elegans* sample. Views at the ground truth in comparison to the PGM matching that was computed after the algorithm for elastic alignment had been applied. See Figure S10 for general explanation. In addition, here a blue connection with two red endpoints indicates that the connection is in the ground truth but not in the PGM matching (false negative). A, B, D, E: the PGM chose additional connections. C: the PGM chose a different connection; rare in *C. elegans*. F, G, H: the PGM decided against connections that are in the ground truth.

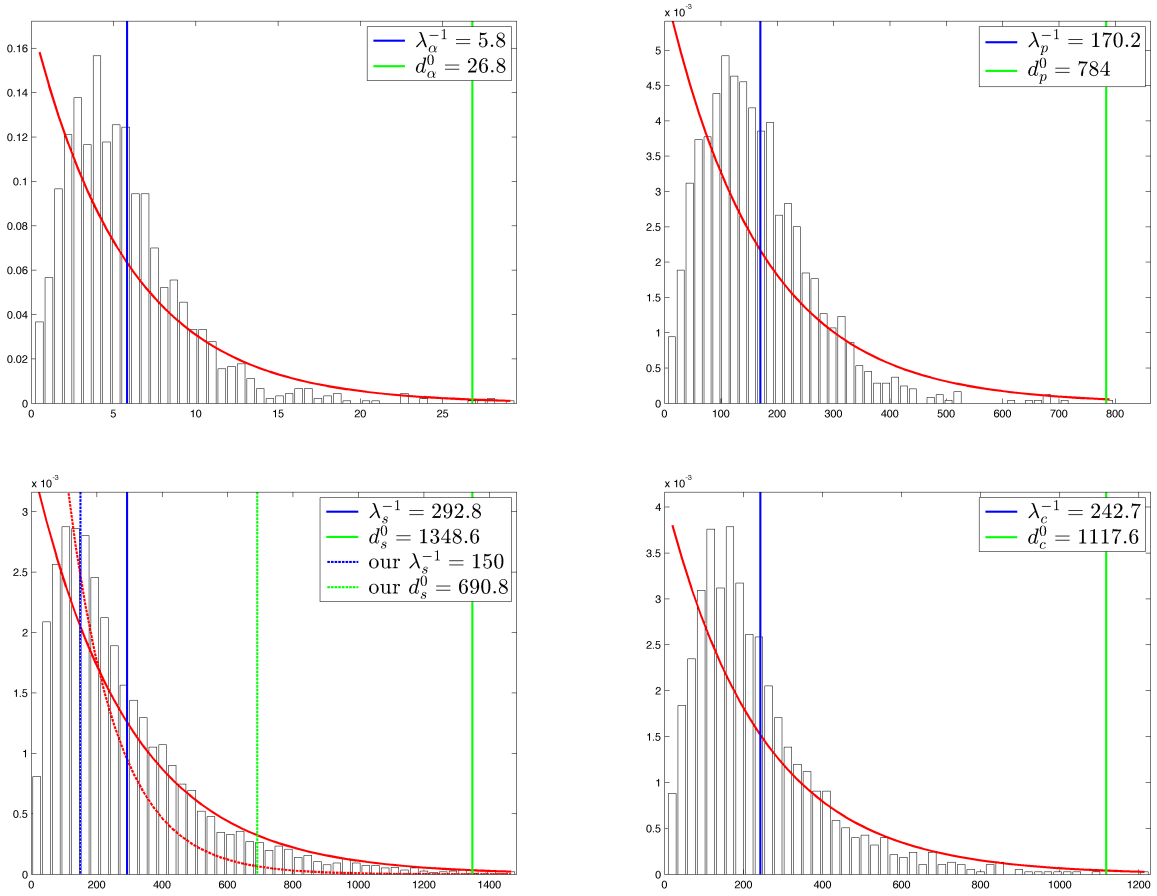

Figure S12. Parameter estimation for PGM factors. Plots show normalized histograms of mutual angle d_α (top left), projected distance d_p (top right), shift difference d_s (bottom left), and direct distance d_c (bottom right) as obtained by analyzing connections that were verified by an expert. Blue vertical lines: Estimated means $\lambda_\alpha^{-1}, \lambda_p^{-1}, \lambda_s^{-1}, \lambda_c^{-1}$. Red curve: corresponding exponential distributions $\lambda \exp(-\lambda x)$. Green vertical line: Placeholder parameters $d_\alpha^0, d_p^0, d_s^0, d_c^0$ computed with a placeholder significance of $r = 1\%$. Dashed lines (bottom left) indicate distribution for our choice of $\lambda_s^{-1} = 150\,\text{Å}$ and corresponding d_s^0.

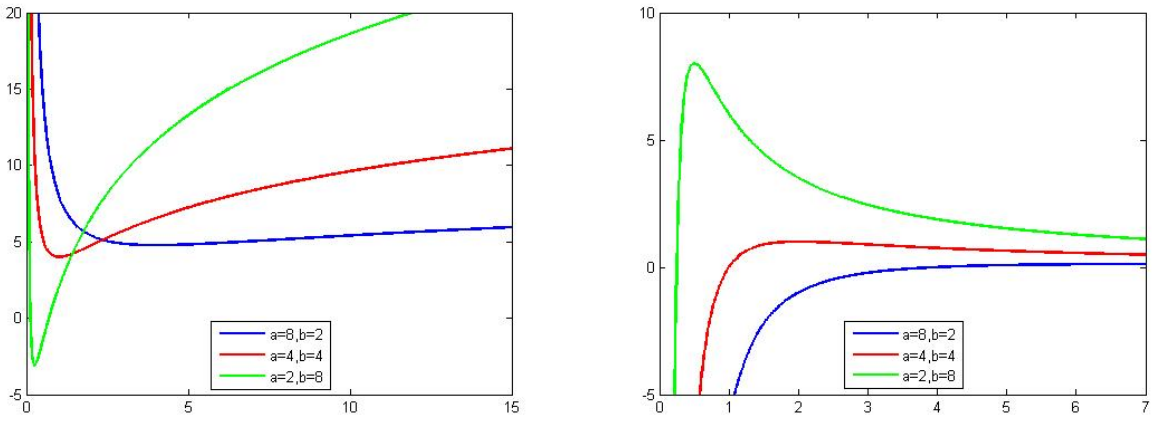

Figure S13. Objective function $Q = a/\sigma^2 + b \log \sigma^2$ plotted against σ^2. Left: the objective function $Q = a/\sigma^2 + b \log \sigma^2$ (Equation 35) plotted against σ^2 for different values of a and b. The smaller the ratio a/b, the sharper the minimum becomes. Right: the derivative $\partial Q/\partial \sigma^2 = -a/\sigma^4 + b/\sigma^2$ plotted against σ^2 for different values of a and b. If σ^2 is initialized on the right side of the peak, the minimum might not be found when optimizing numerically.

$A = \hat{\mathbf{X}}^\top \operatorname{diag}(\mathbf{P}^\top \mathbf{1})\hat{\mathbf{X}}$, $B = \hat{\mathbf{X}}^\top \mathbf{P}^\top \hat{\mathbf{Y}}$, $C = \hat{\mathbf{Y}}^\top \operatorname{diag}(\mathbf{P}\mathbf{1})\hat{\mathbf{Y}}$
$D = \vec{\mathbf{X}}_{2d}^\top \mathbf{P}^\top \vec{\mathbf{Y}}_{2d}$, $E = \vec{\mathbf{X}}_z^\top \mathbf{P}^\top \vec{\mathbf{Y}}_z$

$$
\begin{aligned}
\frac{\partial Q}{\partial s} &= \frac{1}{\sigma^2}(-\operatorname{tr}(B^\top \tilde{R}) + s \operatorname{tr}(C)) \\
\frac{\partial Q}{\partial \rho} &= -\frac{s}{\sigma^2}\operatorname{tr}(B^\top \tilde{R}') - \kappa \operatorname{tr}(D^\top \tilde{R}') \\
\frac{\partial Q}{\partial \kappa} &= -\operatorname{tr}(D^\top \tilde{R}) - \operatorname{tr}(E) - N_P(\frac{1}{\kappa} - \coth(\kappa)) \\
\frac{\partial Q}{\partial (\sigma^2)} &= -\frac{1}{2\sigma^4}(\operatorname{tr}(A) - 2s \cdot \operatorname{tr}(B^\top \tilde{R}) + s^2 \operatorname{tr}(C)) + \frac{N_P}{\sigma^2}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 Q}{\partial s \partial s} &= \frac{1}{\sigma^2}(\operatorname{tr}(C)) \\
\frac{\partial^2 Q}{\partial s \partial \rho} &= -\frac{1}{\sigma^2}(\operatorname{tr}(B^\top \tilde{R}')) \\
\frac{\partial^2 Q}{\partial s \partial \kappa} &= 0 \\
\frac{\partial^2 Q}{\partial s \partial (\sigma^2)} &= -\frac{1}{\sigma^4}(-\operatorname{tr}(B^\top \tilde{R}) + s \operatorname{tr}(C))
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 Q}{\partial \rho \partial \rho} &= \frac{s}{\sigma^2}\operatorname{tr}(B^\top \tilde{R}'') - \kappa \operatorname{tr}(D^\top \tilde{R}'') \\
\frac{\partial^2 Q}{\partial \rho \partial \kappa} &= \operatorname{tr}(D^\top \tilde{R}') \\
\frac{\partial^2 Q}{\partial \rho \partial (\sigma^2)} &= \frac{s}{\sigma^4}(\operatorname{tr}(B^\top \tilde{R}'))
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial^2 Q}{\partial \kappa \partial \kappa} &= -N_P(-\frac{1}{\kappa^2} + \frac{1}{\sinh(\kappa)^2}) \\
\frac{\partial^2 Q}{\partial \kappa \partial (\sigma^2)} &= 0
\end{aligned}
$$

$$
\frac{\partial^2 Q}{\partial (\sigma^2) \partial (\sigma^2)} = \frac{1}{\sigma^6}(\operatorname{tr}(A) - 2s \cdot \operatorname{tr}(B^\top \tilde{R}) + s^2 \operatorname{tr}(C)) - \frac{N_P}{\sigma^4}
$$

Figure S14. Q **derivatives for minimization in algorithm for linear alignment from position and orientation (Figure S2).**