



Article

Untangling Irregular Actin Cytoskeleton Architectures in Tomograms of the Cell with *Struwel Tracer*

Salim Sazzed ¹, Peter Scheible ¹, Jing He ¹ and Willy Wriggers ^{2,*}

¹ Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA; ssazz001@odu.edu (S.S.)

² Department of Mechanical and Aerospace Engineering, Old Dominion University, Norfolk, VA 23529, USA

* Correspondence: wriggers@biomachina.org

Abstract: In this work, we established, validated, and optimized a novel computational framework for tracing arbitrarily oriented actin filaments in cryo-electron tomography maps. Our approach was designed for highly complex intracellular architectures in which a long-range cytoskeleton network extends throughout the cell bodies and protrusions. The irregular organization of the actin network, as well as cryo-electron-tomography-specific noise, missing wedge artifacts, and map dimensions call for a specialized implementation that is both robust and efficient. Our proposed solution, *Struwel Tracer*, accumulates densities along paths of a specific length in various directions, starting from locally determined seed points. The highest-density paths originating from the seed points form short linear candidate filament segments, which are further scrutinized and classified by users via inspection of a novel *pruning map*, which visualizes the likelihood of being a part of longer filaments. The pruned linear candidate filament segments are then iteratively fused into continuous, longer, and curved filaments based on their relative orientations, gap spacings, and extendibility. When applied to the simulated phantom tomograms of a *Dictyostelium discoideum* filopodium under experimental conditions, *Struwel Tracer* demonstrated high efficacy, with F1-scores ranging from 0.85 to 0.90, depending on the noise level. Furthermore, when applied to a previously untraced experimental tomogram of mouse fibroblast lamellipodia, the filaments predicted by *Struwel Tracer* exhibited a good visual agreement with the experimental map. The *Struwel Tracer* framework is highly time efficient and can complete the tracing process in just a few minutes. The source code is publicly available with version 3.2 of the free and open-source *Situs* software package.

Keywords: cryo-electron tomography; segmentation; filament tracing



Citation: Sazzed, S.; Scheible, P.; He, J.; Wriggers, W. Untangling Irregular Actin Cytoskeleton Architectures in Tomograms of the Cell with *Struwel Tracer*. *Int. J. Mol. Sci.* **2023**, *24*, 17183. <https://doi.org/10.3390/ijms242417183>

Academic Editor: Slavica Jonić

Received: 9 October 2023

Revised: 9 November 2023

Accepted: 15 November 2023

Published: 6 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Shape, motility, and transport within a eukaryotic cell are based on an extensive actin cytoskeleton [1]. To visualize actin filaments in their native state, researchers commonly employ cryo-electron tomography (cryo-ET), a specialized imaging technique that enables 3D insight into the internal cellular structure. Cryo-ET has recently shown that actin filaments provide a “missing evolutionary link” between Archaea and complex eukaryotic life forms, such as animals and plants [2]. The cryo-ET by Pilhofer’s lab at ETH Zurich supports an emerging hypothesis that extensive cytoskeletal actin structures arose first in the Asgard Archaea, before the appearance of the first eukaryotes on Earth, and could, therefore, have contributed to the emergence of complex organisms [3]. Moreover, the observed cytoskeletal protrusions in the Archaea suggest a detailed mechanism for eukaryogenesis, in which a primordial Asgard archaeon (the closest known relative of eukaryotes) interacts with the predecessor of the bacterial endosymbiont by means of the actin-powered protrusions and eventually endogenizes it [2,3].

Cryo-ET involves capturing a series of 2D images of a specimen at extremely low temperatures, preserving its frozen state while maintaining its hydrated environment.

This enables detailed imaging of actin filaments without disrupting their native structure. These 2D projections are subsequently aligned using various computational techniques, collectively referred to as 3D reconstruction, to generate the 3D tomogram. The 3–5 nm-resolution tomograms obtained by cryo-ET typically present considerable noise, induced by the limited electron dose used in the image acquisition, and they exhibit directional artifacts from the absence of certain view directions (“missing wedge” in the Fourier space [4]), caused by the limited tilt range of the specimen holder in the microscope.

Determining the organization of actin filaments is highly important as they are responsible for a variety of cellular processes, such as muscle contraction, transport, and cell motility [1], and therefore, it can aid in characterizing experimental, dynamic, or pathological changes in the cytoskeleton [5]. Actin filaments are thin and flexible. They assemble to form diverse organizational patterns, from hexagonal closely packed bundles in the shaft of hair cell stereocilia [6], to directionally biased semi-ordered strands in the stereocilium taper region [4,7], to irregular, randomly oriented actin networks in filopodia [8]. The three types of actin organization can be likened to the appearance of a person’s hair in various stages of tidiness (brushed, windblown, unkempt), and each type requires specialized algorithms for their computational interpretation. Our recent algorithm development was mainly focused on ordered or semi-ordered patterns in stereocilia, where filaments form strands with a mean direction that can be exploited in directional denoising or deconvolution [4,6,7]. In many cells, however, such as in the ancestral Archaea [2], the filaments are randomly oriented and form highly irregular networks, which defy any directionally biased denoising approaches. In this work, we present a novel approach, *Struwwel Tracer*, to quantitatively organize such important disorganized actin networks. (The tool was named after Heinrich Hoffmann’s famously shaggy Struwwel peter character.)

The presence of noise, artifacts, and other structures, such as actin-binding proteins, can obfuscate the filaments and make the tracing difficult. Distinguishing true filaments from noise or other structures in tomograms requires careful examination and expertise, which ultimately make manual tracing of actin filaments a labor-intensive task that requires significant time and effort [8–10]. In addition, manual tracing is often subjective in nature and relies heavily on the annotator’s interpretation. The reliability and reproducibility of the annotation procedure are greatly compromised by factors such as low contrast, various types of artifacts, and inherent uncertainty. This is particularly evident when the procedure is applied directly to whole-cell tomograms (i.e., without any subtomogram averaging) with typical resolutions below 2 nm and varying in orientation due to the missing wedge effect. Moreover, when dealing with large datasets containing 2D image stacks, the repetitive nature of the annotation task increases the chance of error, such as an inadvertent skipping between neighboring filaments. To mitigate these potential errors and to make the tracing objective and reproducible, it becomes necessary to develop automated approaches that can be independently validated.

Over the years, a number of automatic methods have been developed to trace actin filaments [4,8,10–13]; however, the existing methods have certain limitations:

1. A significant portion of these approaches can be ruled out for the present work because they are only applicable to non-actin filaments such as microtubules [14] and polysaccharides [15], or to actin filaments observed in relatively clean light or confocal microscopy images [16,17].

2. Among the cryo-ET-related actin tracing methods, some are only applicable for the tracing of well-ordered filaments [4,6,7]. Among these, one noteworthy tool is our *Spaghetti Tracer* approach [7]. *Spaghetti Tracer* introduced a paradigm shift in the tracing of semi-regular actin filaments because it is a dynamic-programming-based method at the voxel level that does not require an expensive missing wedge correction, template convolution, or deconvolution. Therefore, it yields a substantial improvement in time efficiency over template convolution [8,10] or deconvolution methods [4], enabling fast and accurate tracing of such filament arrangements. (The accuracy of *Spaghetti Tracer* was validated in a rigorous statistical analysis, achieving F1-scores of 0.86–0.95 on phantom tomograms under experimental conditions.) The success of *Spaghetti Tracer* motivated us to extend its capabilities to randomly oriented actin filaments in the present work.
3. There are very few earlier algorithms that are agnostic of the relative orientations and distances of the actin filaments, so that they can trace central lines of irregular filaments individually without leveraging the information of adjacent filaments or requiring a mean direction. *Volume Tracer* [8] utilized an expensive genetic-algorithm-based search employing a population of cylindrical templates (combined with a bi-directional tracing) to detect randomly oriented filaments in *Dictyostelium discoideum* filopodia. The genetic algorithm was implemented as part of our group's free, open-source *Situs* and *Sculptor* packages, but it required extensive computational time on the order of several days when applied to a complete tomogram, without guaranteeing convergence (leading to false negatives when a user is impatient). Co-author Rigort also developed a similar template-matching method independently [10]. This approach was implemented in *Amira*, a commercial software requiring a paid license and limiting any algorithmic modifications by third parties or end users.
4. Recently, a number of deep-learning-based approaches have been proposed for the segmentation of diverse biological assemblies, including actin [18,19]. For example, Chen et al. [18] presented a deep-learning-based segmentation approach for a voxel-level classification of shapes of interest in the tomogram and integrated the approach into the *EMAN2* [20] package. However, these segmentation tools are generic in nature and are not specifically designed for filamentous shape structures. Besides, they require users to annotate training data and fine-tune the deep learning model, which could be a laborious process. These segmentation approaches only provide a voxel-level density segmentation, without any tracing of central lines. Recent studies that used these segmentation methods subsequently required separate approaches, such as the above template matching, for the additional tracing [11–13].

In summary, the current state-of-the-art of identifying actin filament networks in cryo-ET has several drawbacks, including the fragmentation of tools and packages, the prerequisite to buy commercial software (*Amira*), complex manual interventions (e.g., manual annotation for segmentation or the training of deep learning networks), and computational expense. Our newly proposed approach, demonstrated in Section 2, can address all these drawbacks. *Struwel Tracer* is an efficient, accurate, free, open-source tool for tracing randomly oriented filaments in actin networks (Figure 1). We describe in Section 3 how the framework first detects local seed points (Figure 1) in the map, from which it generates short candidate filament segments (CFSs). Next, path densities are accumulated by exploring all potential filament paths within 45° search pyramids from the *x*, *y*, and *z* directions. After placing CFSs across the map based on the maxima of path densities, we describe in Section 3 how a manual segmentation can be efficiently performed by visualizing an intermediate “pruning map” in a third-party molecular graphics program. Finally, the surviving CFSs are iteratively fused (Figure 1) into longer, curved filaments based on their relative orientations and gap spacings after extension.

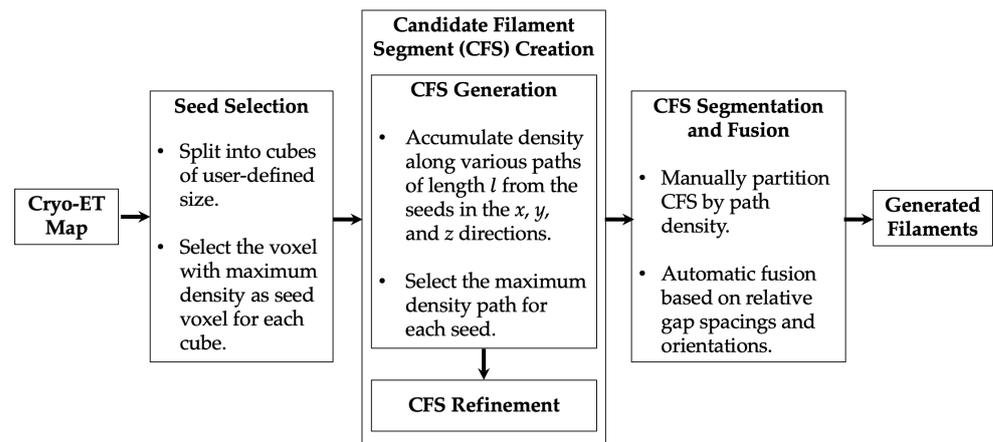


Figure 1. Overall flow diagram of *Struwwel Tracer* as described in Section 3.

2. Results and Discussion

To evaluate the performance of *Struwwel Tracer*, a comprehensive statistical F1-score analysis was performed on simulated tomograms of a *Dictyostelium discoideum* filopodium with a known ground truth. Different levels of noise were added to simulate realistic imaging conditions. Additionally, the effectiveness of *Struwwel Tracer* was demonstrated on an experimental tomogram of a mouse fibroblast lamellipodium that was interpreted for the first time by filament tracing. Finally, we report the implementation details and computation times of our software.

2.1. Statistical Evaluation of Tracing Accuracy in Simulated Tomograms

In earlier work [7], we established a rigorous statistical evaluation protocol [5] for testing the accuracy of filament-tracing approaches. This protocol is based on simulated phantom tomograms that we generated from known filament traces under realistic conditions that matched the noise profile and missing wedge properties of an experimental tomogram (see Section 3.1). To evaluate *Struwwel Tracer*, we computed the precision, recall, and F1-scores based on the observed agreement.

The ground truth filament voxels of simulated *Dictyostelium discoideum* filopodium maps (see Section 3.1) were compared with the filament voxels predicted by the tracing. Due to the inherent high noise levels present in cryo-ET images, previous studies [5] have highlighted the limitations of conducting one-to-one voxel-level comparisons between the ground truth and predictions. Therefore, here, we used a more-pragmatic approach; rather than direct voxel-level comparison, we performed the comparison based on a neighborhood range [5,7] of three voxels. The true positive (TP), false positive (FP), and false negative (FN) voxels were then computed in the following way:

True positive: If a ground truth filament voxel is found within a $3 \times 3 \times 3$ -voxel neighborhood of a predicted voxel, it is considered a TP.

False positive: If no ground truth voxel is found within a $3 \times 3 \times 3$ -voxel neighborhood of a predicted voxel, it is considered an FP.

False negative: If no predicted voxel is found within a $3 \times 3 \times 3$ -voxel neighborhood of the ground truth voxel, it is considered an FN.

The recall (R), precision (P), and their harmonic mean (F1) are then computed according to

$$R = \frac{TP}{TP + FN} \quad (1)$$

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$F1 = \frac{2 * R * P}{R + P} \quad (3)$$

Table 1 presents the precision, recall, and F1-scores obtained by *Struwwel Tracer* in simulated tomograms with varied levels of noise (see Section 3.1). *Struwwel Tracer* achieved a high F1-score of 0.90 when applied to the lowest-noise map we tested. The high precision score of 0.97 indicated that the framework recognized mostly true filaments in the simulated tomogram (FP predictions were negligible). As the noise level increased, the performance degraded a bit due to the slightly lower recall values (i.e., a small number of true filaments were missed); however, the obtained F1-scores still remained above 0.8. Given the inherent noise in the tomogram, the recall scores ranged slightly below the precision scores since missing filaments' densities resulted in FNs.

Nevertheless, the observed F1-scores ranging from 0.85 to 0.90 (illustrated in Figure 2 for noise levels of 0.35 to 0.95) presented a remarkable level of accuracy for a density-based structure prediction. For comparison, we achieved much lower F1-scores of 0.72 for alpha helices and 0.65 for beta sheets in a recent state-of-the-art deep learning prediction of secondary structure features in cryo-electron microscopy maps [21]. The high *Struwwel Tracer* F1-scores were only slightly below those observed earlier with *Spaghetti Tracer* (0.86–0.95) on much-more-ordered actin filaments [7].

Table 1. A performance comparison of the proposed *Struwwel Tracer* approach for simulated tomograms of a *Dictyostelium discoideum* filopodium (see Section 3.1) at various noise levels.

| Noise Level | Precision | Recall | F1-Score |
|-------------|-----------|--------|----------|
| 0.35 | 0.97 | 0.85 | 0.90 |
| 0.50 | 0.97 | 0.81 | 0.88 |
| 0.65 | 0.96 | 0.84 | 0.89 |
| 0.80 | 0.95 | 0.79 | 0.87 |
| 0.95 | 0.95 | 0.78 | 0.85 |

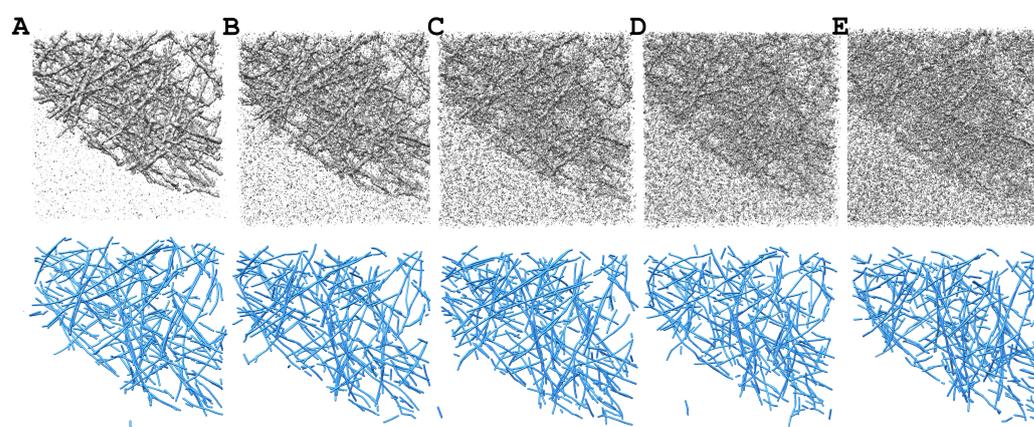


Figure 2. The simulated tomograms (top row) used in this study (isocontour level: mean + 1.5 × standard deviation) and the corresponding *Struwwel Tracer* predictions (bottom row). The simulated tomograms are shown at the following noise levels (see Section 3.1): (A) 0.35, (B) 0.50, (C) 0.65, (D) 0.80, and (E) 0.95. All molecular graphics in this work were created with UCSF Chimera [22].

2.2. Measuring Filament Center Lines in a Previously Untraced Experimental Tomogram

We applied *Struwwel Tracer* to an experimental tomogram of a mouse fibroblast lamellipodium [23], which was deposited by the authors in the Electron Microscopy Data Bank (EMDB) [24] as EMD-11870. This specific map has not been computationally traced due to their focus on Arp2/3 protein branch junctions in [23], so a measurement of actin center lines could provide complementary information to the paper. (Tracings were shown for similar lamellipodia tomograms in [11], but only EMD-11870 is publicly available.)

Since this was a previously untraced map and we did not have any ground truth (i.e., no annotated model), we provide in Figure 3 a visual comparison to demonstrate the

excellent agreement between the density and the predicted filament center lines. EMD-11870 [23] is a larger tomogram that contains randomly oriented and branched actin filaments. For our demonstration in Figure 3A, we selected a $312 \times 320 \times 20$ sub-region of the tomogram in which filamentous patterns of actin can be clearly appreciated. The *Struwvel Tracer* prediction results are illustrated in Figure 3B,C.

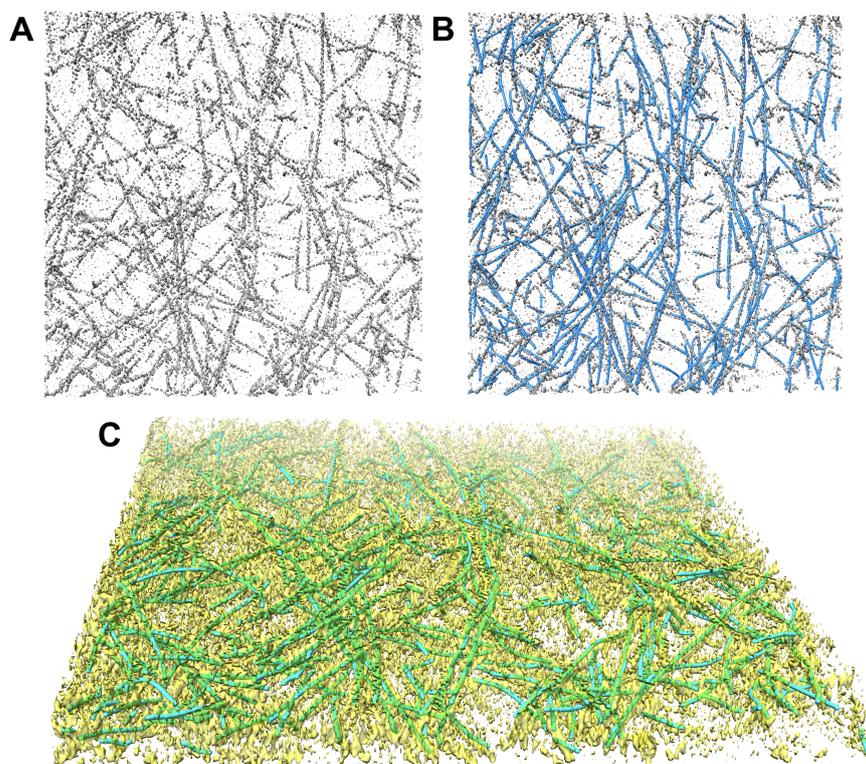


Figure 3. (A) A $312 \times 320 \times 20$ subtomogram cropped from EMD-11870 [23] map indices $i = 100\text{--}411$, $j = 200\text{--}519$, and $k = 74\text{--}93$. The isocontour level in the rendering is 0.4518 (mean + $1.5 \times$ standard deviation). (B) The map density in (A) overlaid by the *Struwvel Tracer*-predicted filament center lines (blue). (C) The 3D perspective rendering of the map density in (A) (yellow) overlaid by the *Struwvel Tracer*-predicted filament center lines (green).

2.3. Computation Time and Manual Intervention

In terms of computational efficiency, *Struwvel Tracer* demonstrated outstanding performance, surpassing the earlier tracing method [8] (which took days of computational time) by several orders of magnitude in speed. On an Apple MacBook Pro equipped with a 2.6-GHz Intel Core i7 processor, we observed that, for a simulated tomogram of size $200 \times 200 \times 71$ voxels, *Struwvel Tracer* took approximately three minutes of runtime. Given the ongoing developments in cryo-ET, it is expected that hundreds of tomograms can soon be acquired within a few days [11]. *Struwvel Tracer* would be able to match this processing speed.

In addition, the proposed method does not require extensive manual intervention, which is the case for interactive and deep-learning-based tools for which users either need to tune many parameters or label training data and train the model for prediction. For the *Struwvel Tracer* approach, user intervention is needed only once in the CFS segmentation step to select an approximate value of the threshold in the pruning map. This requires opening the pruning map in visualization software and takes at most a few minutes.

2.4. Software Implementation and Dissemination

The newly developed *Struwvel Tracer* approach is set to be seamlessly integrated into version 3.2 of *Situs*, a widely used, open-source software package for biological image

interpretation (see the Data Availability Statement). *Struwvel Tracer* will be disseminated with *Situs* Version 3.2 as a new command line tool named *strwtrc* (following the seven-letter naming convention). To ensure full compatibility with *Situs*, *strwtrc* was implemented using C/C++, the primary language of *Situs*. A notable advantage of *strwtrc* is that it does not rely on any third-party libraries, making it self-contained and efficient. Moreover, *strwtrc* is fully compatible with both the Linux and MacOS operating systems.

In addition, *strwtrc* was designed to be user friendly and accessible, requiring minimal programming language proficiency and no prior coding experience. We intentionally kept the number of user-defined parameters to a minimum (Table 2), making it easier for users to operate the software. The default parameters were designed to work well for most cases; however, it is important to note that they may not be optimal for every possible dataset. A summary of all parameters is shown in Table 2. A comprehensive user manual and tutorial will be included with the *Situs* package to guide users through the individual stages of the approach.

Table 2. Command line parameters of the *strwtrc* program implemented in *Situs* [8] and corresponding stages of the approach (see Sections 3.2–3.5 and the Data Availability Statement).

| Parameter Name | <i>strwtrc</i> Argument | Description | Default Value | Program Stages |
|----------------------------|-------------------------|---|---|--|
| Required Parameter | | | | |
| Threshold | -thr | Threshold for partitioning the CFS by the normalized path density. | N/A (user-defined based on the pruning map) | CFS segmentation |
| Optional Parameters | | | | |
| Length | -len | Length (infinity norm) of the CFS in voxel units. Internally, this also defines the spacing of the cubic grid for placing seed points (half this value; see the text) and the extension length of the CFS (same value). | 10 | Automatic seed selection, CFS generation, and CFS fusion |
| Gap Spacing | -gap | Maximum gap spacing, in voxels, tolerated while fusing adjacent CFSs. | 10 | CFS fusion |
| Fusion Angle | -ang | Maximum angle, in degrees, tolerated while fusing adjacent CFSs. | 30 | CFS fusion |

3. Materials and Methods

Filaments in an irregular actin network, such as those found in filopodia, lamellipodia, or stress fibers, are particularly challenging to trace due to their arbitrary orientation and potential overlap or branching with other filaments. We report in the following how simulated phantom tomograms were generated from ground truth traces to provide a controlled basis for our method development and validation. Subsequently, we describe the stages of the proposed tracing framework, shown in Figure 1, to address such challenges in the tracing of actin filaments with arbitrary orientations.

3.1. Simulated Phantom Tomograms

In our previous work [25], we introduced *TomoSim*, a tomogram simulation approach that leverages pre-existing filament traces to generate authentic phantom tomograms, whose noise color, strength, and missing wedge were matched to experimental tomograms. In this study, we extended our simulation method to address the random orientations of the filaments.

To establish a reliable benchmark for validating *Struwvel Tracer*, we utilized actin traces of a *Dictyostelium discoideum* filopodium. A spline interpolated from the filament trace in Supplementary Data 5 from Rusu et al. [8] was volumized using a Gaussian filter with a full width at half maximum of 9 nm with a voxel spacing of 1.912 nm.

As elaborated in our previous work [25], the simulation of a tomogram aims to replicate the noise and missing wedge artifacts typically encountered in experimental maps. The simulated tomogram does not incorporate non-filamentous biological features, such as membranes. Consequently, the simulated tomogram serves as a well-defined ground truth specifically tailored to assess the accuracy of any filament-tracing framework. We used an experimental tomogram obtained from Supplementary Data 2 from Rusu et al. [8] for noise-color modeling and signal-to-noise ratio (SNR) calibration. A wedge was masked in the frequency domain to emulate a tilt range of $\pm 45^\circ$.

Using the color-matched noise profile, we matched the signal strength to the experimental tomogram's SNR. We used this as the basis for testing our tracing against a range of noise levels. Noise levels were quantified as the noise intensity ratio relative to the experimental tomogram. For example, a noise level of 0.5 would have half the noise intensity of the experimental tomogram and twice the SNR. The experimental map we used for the calibration was the raw tomogram prior to any denoising or local normalization (which is typically performed on a tomogram prior to tracing [8]). Therefore, a noise level of 1.0 is a worst-case scenario that is not encountered in tracing practice. To simulate the effect of the missing denoising step, we generated a range of five equally spaced noise levels from 0.35 to 0.95, as normalized by the worst-case level. (We did not calibrate our noise on the processed tomogram [8] to avoid any subjective bias from that study and, instead, used a range of reduced noise levels as calibrated from the raw tomogram.) Note that the absolute noise scale in simulated tomograms is not strictly defined anyway because of the hidden misalignment of unknown true filament densities with the prescribed ground truth traces [25]. (So, the relative noise strengths are more relevant than the absolute scale).

3.2. Automatic Seed Selection

The detection of filaments in cellular tomograms is initiated from suitably chosen high-density seed points. Manual filament tracing relies on user-provided seed points, which were considered as the ground truth in previous work [6]. However, due to the labor-intensive tagging, such manual seed point selection is relatively sparse and introduces uncertainty and a subjective bias, which we wished to avoid. An automated approach is less concerned with the expense of the selection and can consider a denser oversampling of potential seed points, but it might introduce false positives. Therefore, instead of considering seed points as the definitive ground truth, we utilized them solely as starting points for the CFSs, which will be screened and refined later.

For an automatic seed point selection on a 3D grid, we identified high-density voxels within a local neighborhood. To achieve this, the tomogram was sub-partitioned into non-overlapping 3D cubes along each axis. The voxel with the highest density value within each cube was then designated as a seed voxel. In semi-regular filaments [7], the cube side length was chosen to be identical to the CFS length l in the CFS creation stage (below), which provides a natural length scale for the coarse-grained seed placement. However, in irregular actin networks, we expected the final seeds to be more irregularly distributed in the search cubes, so a denser sampling with a grid spacing of $l/2$ was implemented in *Struwvel Tracer*. For this research, we used a CFS length of $l = 10$ voxels, resulting in seed search cubes of dimensions $5 \times 5 \times 5$ voxels. (At a voxel spacing of 1.912 nm in the phantom tomograms above, five voxels correspond approximately to the 9 nm actin filament diameter, so the seed point density should afford a complete detection of filaments.)

We assumed that seed points with local high density are necessary, but not sufficient prerequisites for true filaments. Identifying true seed voxels (that are part of a true filament) is difficult at this stage. This difficulty arises from the low SNR and the presence of missing wedge artifacts or other structures in tomograms. The determination of whether a density segment that originates in a seed belongs to a filament or not is performed in a later stage of the approach. Consequently, we refer to the short filament segments originating from a seed voxel not as predicted filament segments, but as *candidate* filament segments (CFSs).

3.3. CFS Creation

This stage of the approach (Figure 1) comprises two sub-steps, CFS generation and CFS refinement.

3.3.1. CFS Generation

A CFS originates in the seed voxel (i, j, k) and terminates in the end voxel (i', j', k') , which is determined for each seed (i, j, k) by the following equations. The length l of the CFS in this study is user defined and is equivalent to the infinity norm of the resulting CFS vector, $l = \max\{i' - i, j' - j, k' - k\}$. As mentioned previously, we used a value of $l = 10$ voxels in this paper.

On a cubic grid, let the local density at each voxel D be globally normalized to a range of 0 to 1. Three separate path densities, denoted as $PD_{x,y,z}$, are initialized at the seed voxel (i, j, k) for the three Cartesian $x, y,$ and z axes:

$$PD_{x,y,z}(i, j, k) = D(i, j, k) \tag{4}$$

These path densities are then accumulated for each Cartesian axis over l voxels in the forward (positive) direction within the search pyramids that have the seed (i, j, k) as the vertex and can deviate by up to 45° from the corresponding axis. The search pyramids, shown in green (x), red (y), and blue (z) in Figure 4, will be formally defined below. The pyramidal search approach was inspired by the *Spaghetti Tracer* algorithm [7]. However, unlike the earlier method, which was focused on a single dominant direction, we made no assumption about CFS directionality in the present work. Therefore, three Cartesian search pyramids will be combined in the following, to cover all possible orientations.

At each intermediate voxel (i'', j'', k'') within a search pyramid, the corresponding path density (initialized in Equation (4)) is then accumulated from up to nine contributing neighbor voxels from a previous cross-section. To illustrate this, when tracing in the y direction (red in Figure 4), the path density $PD_y(i'', j'', k'')$ is accumulated from the preceding y -slice, $j'' - 1$ by mathematical induction. The accumulation scheme is applied to all intermediate voxels within the search pyramids:

$$PD_y(i'', j'', k'') = D(i'', j'', k'') + \max_{\substack{m,n \in \{-1,0,1\} \\ \text{(if contributing)}}} PD_y(i'' + m, j'' - 1, k'' + n). \tag{5}$$

Note that only neighbors $m, n \in \{-1, 0, 1\}$ within the search pyramid are considered. Therefore, as described in [7], the accumulation zone from the seed (i, j, k) to the final endpoint (i', j', k') is actually an intersection of two overlapping pyramids, i.e., path densities are accumulated only in a relatively localized region. Similarly, $PD_x(i'', j'', k'')$ and $PD_z(i'', j'', k'')$ are accumulated in their respective directions, with indices permuted accordingly:

$$PD_x(i'', j'', k'') = D(i'', j'', k'') + \max_{\substack{m,n \in \{-1,0,1\} \\ \text{(if contributing)}}} PD_x(i'' - 1, j'' + m, k'' + n), \tag{6}$$

$$PD_z(i'', j'', k'') = D(i'', j'', k'') + \max_{\substack{m,n \in \{-1,0,1\} \\ \text{(if contributing)}}} PD_z(i'' - m, j'' + n, k'' - 1). \tag{7}$$

The final forward path densities for the green (x), red (y), and blue (z) search pyramids in Figure 4 are then computed on the forward-facing square bases of the pyramids:

$$FPD_x(i, j, k; l) = \max_{\substack{j-l \leq j' \leq j+l \\ k-l \leq k' \leq k+l}} PD_x(i+l, j', k'), \tag{8}$$

$$FPD_y(i, j, k; l) = \max_{\substack{i-l \leq i' \leq i+l \\ k-l \leq k' \leq k+l}} PD_y(i', j+l, k'), \tag{9}$$

$$FPD_z(i, j, k; l) = \max_{\substack{i-l \leq i' \leq i+l \\ j-l \leq j' \leq j+l}} PD_z(i', j', k+l). \tag{10}$$

The $FPD_{\{x,y,z\}}(i, j, k; l)$ values in Equations (8)–(10) range from 0 to $l + 1$ (number of voxels in the CFS). For the segmentation in the next stage of the algorithm, we divided the maximum between the three FPD values by $l + 1$ to obtain a single normalized NPD value, which ranges from 0 to 1:

$$NPD(i, j, k; l) = \frac{\max_{u \in \{x,y,z\}} FPD_u(i, j, k; l)}{l + 1}. \tag{11}$$

The combined 3D search region (roughly a hemisphere of directions comprising the three search pyramids in Figure 4) accounts for all possible (unsigned) filament orientations. The final endpoint (i', j', k') of the CFS is then defined as the point that exhibits the maximum FPD_u , according to the winning u axis that defines the $NPD(i, j, k; l)$ in Equation (11).

The path-based density accumulation addresses the challenge posed by the substantial noise levels in the tomogram. Rather than solely focusing on the density of individual voxels or on density measurements along predetermined axes, the algorithm aims to detect an elongated high-density path with a specific infinity norm l and an orientation determined by the maximum attainable path density.

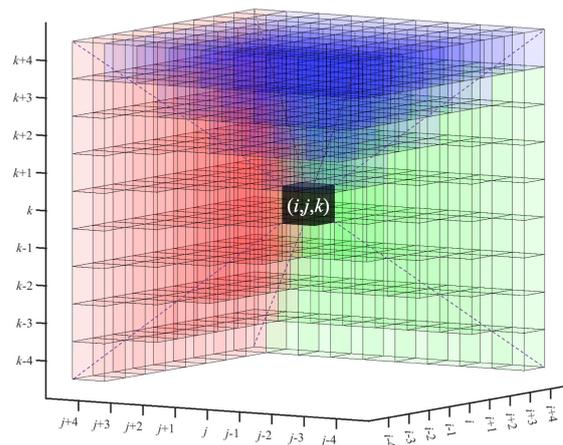


Figure 4. Pyramidal density search regions along different axes based on Equation (8) (green), Equation (9) (red), and Equation (10) (blue), starting from the seed voxel (black) at position (i, j, k) . For a clearer illustration, we used a CFS length (infinity norm) $l = 4$ voxels for the search pyramid dimensions, whereas $l = 10$ was used elsewhere in this paper.

3.3.2. CFS Refinement

The forward CFSs were subjected to an additional screening step based on backward tracing. From the end point (i', j', k') of each forward CFS, a backward CFS with new end points (i^*, j^*, k^*) was constructed by applying the above tracing algorithm in reverse. It was expected that, for a high-density CFS that is part of a true filament, the backward

tracing will follow a comparable path in the reverse. Conversely, forward and backward CFS characterized by low density or noise, which are unlikely to be part of any filament, would diverge from each other. To determine the similarity of paths between forward and backward tracing, the angle between the points (i, j, k) , (i', j', k') , and (i^*, j^*, k^*) was calculated, and a fixed angle threshold of 20° was employed to screen out inconsistent forward and backward CFSs. An example of the intermediate results after the refinement stage is shown in Figure 5. The surviving CFSs (Figure 5B) exhibited the preferred orientation of filaments in the actin-rich interior of the cell, whereas the exterior of the cell (lower left) yielded unstructured CFS patterns, which were not true filaments and will need to be filtered out in the following.

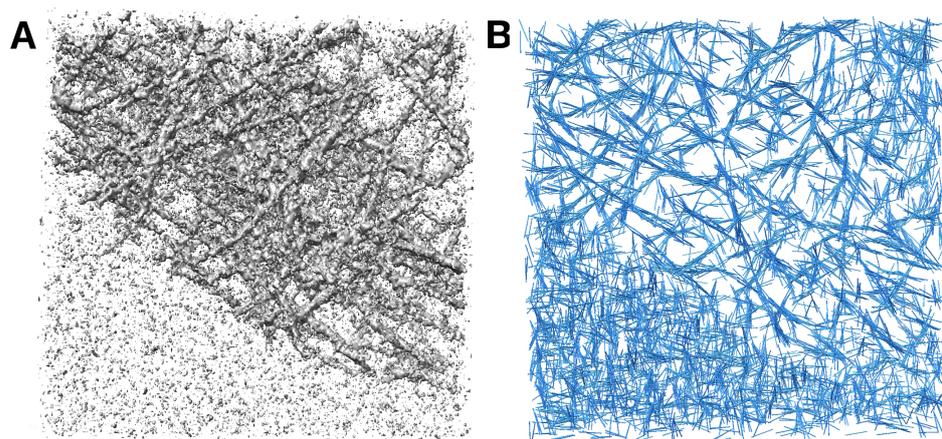


Figure 5. (A) Density map of the simulated *Dictyostelium discoideum* filopodium tomogram at a noise level of 0.50 (shown at an isocontour level of mean + $1.5 \times$ standard deviation). Note that the cell membrane was not simulated. (B) Filament segments of length $l = 10$ voxels, generated by applying the forward-path-based density accumulation and followed by backward tracing refinement.

3.4. CFS Segmentation

Figure 5B illustrates the segmentation problem in the irregular actin networks, which are the focus of this work. In earlier work on ordered actin filament bundles (with a dominant orientation), we found that it was possible to screen out spurious CFSs (such as in the lower left corner) using an automated binning method that finds a suitable *NPD* threshold above which the CFSs remain contained in the true filament region. CFSs with *NPD* values below this threshold were then eliminated automatically. We propose here to use a similar *NPD* threshold segmentation. However, irregular actin network CFSs exhibit low *NPD* contrast (there was no separate directional denoising stage as in regular filaments [7]), and we did not assume that irregular networks are localized in a specific region. Therefore, it would be desirable to fine-tune the threshold as a continuous parameter instead of prescribing discrete bin values. We implemented a novel way to allow users to perform such a continuous thresholding quickly and conveniently with a third-party molecular graphics program such as VMD [26], Sculptor [8] or UCSF Chimera [22]. Our approach outputs a so-called “pruning map”, which contains the *NPD* values masked by the CFS locations. Voxels in close proximity (i.e., one voxel) to the CFSs are assigned the *NPD* value (Equation (11)) of the CFS. When a voxel can draw *NPD* values from multiple CFSs, the highest value is retained.

In Figure 6, the pruning map returned by *Struwel Tracer* is shown at different isocontour levels. By adjusting the level in the histogram window, users can visually determine the threshold in the pruning map (i.e., the threshold *NPD*) that represents likely filament segments. An overly low threshold would result in numerous false positives (Figure 6A), whereas an overly high threshold would lead to a significant number of false negatives (Figure 6E). Visual examination indicates that the isocontour values within the range depicted in Figure 6C,D are likely to represent the filaments accurately. Note that the

subsequent steps of the algorithm are not very sensitive to the selected threshold value because of the screening and gap filling that still follow. The main purpose of the visual segmentation was to suppress spurious filaments, as is shown in Figure 6A, so a reasonable guess of the threshold parameter suffices in practice.

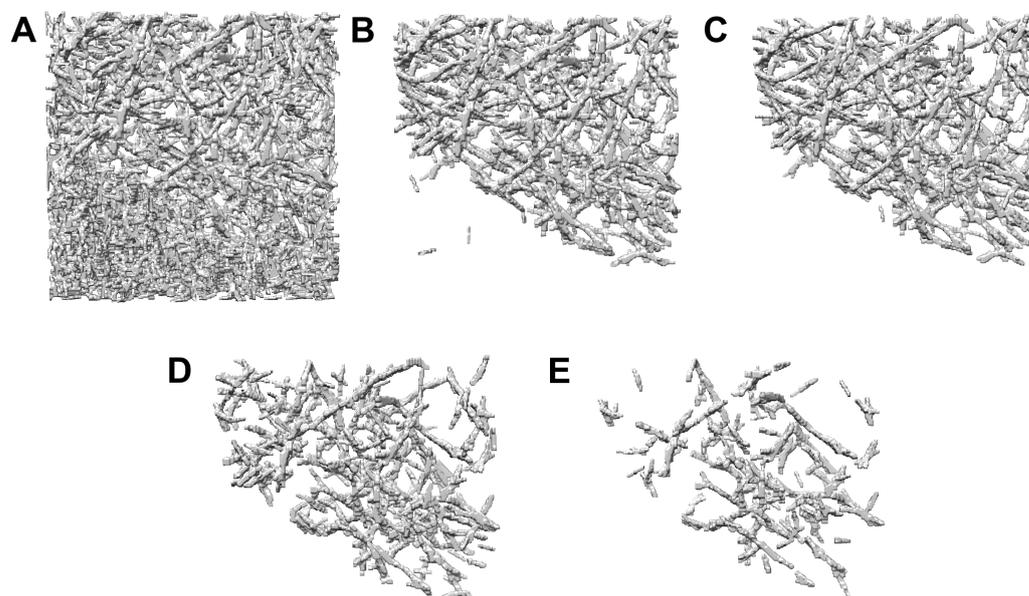


Figure 6. Pruning map of the simulated tomogram in Figure 5A (noise level: 0.50) at different isocontour levels: (A) 0.40, (B) 0.45, (C) 0.50, (D) 0.55, and (E) 0.60.

3.5. CFS Fusion

In the filament-fusion stage, we employed multiple geometric strategies to combine short, linear CFSs (Figure 5B) into curved, longer filaments. To account for the irregular organization of actin networks, the following fusion steps were specifically designed for *Struwvel Tracer* and were different from the collinearity test we developed earlier for oriented filaments [7].

3.5.1. Fusion Based on Physical Proximity

The initial fusion step considers the relative spacing and orientation of neighboring CFSs. Adjacent filaments that exhibit similar orientations (default angle tolerance of CFS center lines: 30°) will be connected. We determined if two adjacent CFSs overlap or touch each other based on their spacing along the center lines of the CFSs (the default gap tolerance was 10 voxels). CFSs that meet the criteria are combined by connecting the end point of a CFS to the starting point of the successive CFS. Since it is not required that two CFSs exhibit perfectly matched orientations and positions before they are merged, we smoothed the center line of the fused CFSs by sampling, interpolation, and the removal of redundant points.

3.5.2. Fusion by Extension

In addition to the proximity fusion of short CFSs, we tested whether a one-time extension of CFSs by a length l can extrapolate CFSs such that they make contact with another CFS using the same angle and spacing criteria. The extension step aimed to fill up any noise-induced gaps present in the filaments. Due to the high noise levels and missing wedge artifacts in the tomograms, it is possible that true filaments exhibit regions of density at the level of the noise, which may be excluded by a high threshold during the visual inspection of the pruning map (Figure 6). We, therefore, examined whether the one-time extension by a length l can help to fuse more CFSs. Similar to the previous step,

we smoothed the center lines of fused CFSs by sampling, interpolation, and the removal of redundant points.

4. Conclusions

The release of *Struwvel Tracer* is the capstone of several years of development effort on actin filament tracing that extends and completes our set of efficient solutions for hexagonal, closely packed bundles (*BundleTrac* [6]) and semi-regular bundles with a dominant direction (*Spaghetti Tracer* [7]) to irregular networks of randomly oriented actin filaments. *Struwvel Tracer* also completes the paradigm shift we began with *Spaghetti Tracer*, when we first used a dynamic-programming-based method at the voxel level that does not require an expensive missing wedge correction, template convolution, or deconvolution. Therefore, both *Spaghetti Tracer* and *Struwvel Tracer* yield a substantial improvement in time efficiency over earlier template convolution or deconvolution approaches. This development was guided by a rigorous optimization of the performance using a statistical F1-score analysis enabled by simulated phantom tomograms.

The proposed framework incorporates a directional path search in all three Cartesian directions (x , y , and z) to ensure a comprehensive coverage of filament orientations. The algorithm identifies the path with the highest density, generating short filament segments, which are subsequently combined to form the final filament traces. The evaluation using F1-scores on simulated tomograms proved the algorithm's high efficacy in filament tracing. Visual inspection of the results further confirmed its agreement with experimental tomograms. Our approach is robust and fast, works with simple parameter settings, and can deliver a comprehensive performance on simulated and experimental datasets. Moreover, in contrast to deep-learning-based approaches that rely on substantial processing power, such as dedicated GPUs, the implementation in the *Situs* package does not necessitate such resources. It can seamlessly operate on any conventional laptop or desktop computer with a standard CPU.

In addition to the high efficacy in tracing, the proposed framework is also robust as it is, in principle, capable of tracing various types of filaments, whether they are randomly oriented networks or regular (with a dominant direction). We note, however, that our earlier tools (*Volume Tracer* [8], *BundleTrac* [6], *ConDe* [4], and *Spaghetti Tracer* [7]) have unique features that were optimized for their specific applications (e.g., an additional capability to detect alpha helices in *Volume Tracer*, the use of hexagonal bundle templates in *BundleTrac*, the ability to prescribe diverse shape templates in *ConDe*, and an integrated directional denoising that takes advantage of an ordered filament arrangement in *Spaghetti Tracer*). A comprehensive F1-score comparison of various tools and conditions could be performed in a future review.

As the study of the actin cytoskeleton is of increasing importance in biology, the proposed tool will become useful to structural biologists in need of free, open-source software solutions. Although this manuscript focused primarily on the computational aspects of actin filament tracing, the developed framework can be applied by any experimental lab to newly acquired experimental tomograms of the actin cytoskeleton. Functionality for measuring actin filament length distributions and the angles between filaments, and for labeling branch junctions [10,23], can be included in future work.

Author Contributions: S.S., W.W. and J.H. participated in the methodology development of *Struwvel Tracer*. P.S. and W.W. participated in data simulation. S.S. implemented the approach and performed the tests. W.W. and S.S. wrote the manuscript, and all authors edited it. All authors have read and agreed to the published version of the manuscript.

Funding: The work in this article was supported in part by the Frank Batten Endowment at Old Dominion University and by NIH R01-GM062968.

Data Availability Statement: The source of *Struwvel Tracer* can be freely downloaded at <https://situs.biomachina.org> in *Situs* version 3.2.

Acknowledgments: This article is dedicated to the memory of Junha Song, born 4 January 1994 in Seoul, Republic of Korea, who unexpectedly left us on 16 February 2023. Junha is survived by his father, Youngkyu Song; his mother, Kyeongsun Yeo; and his brother, Kee Young Song. As an undergraduate student at UC Berkeley and a member of Manfred Auer's lab at Lawrence Berkeley National Laboratory, Junha co-authored several publications with us on actin filament tracing [4,6,9]. Between 2017 and 2019, he and Auer participated in weekly online meetings with our group during the early stages of this project, and we are grateful to both of them for many helpful discussions. Junha was a master of visualizing complex molecular structures and created manual tracings of tomograms we used as ground truths for method validation. Junha inspired us to continue this work while he had returned to Seoul and was preparing to attend graduate school in the U.S. Junha had an immense passion for biophysics, and we know that he would have continued to excel in his research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Schoenenberger, C.A.; Mannherz, H.G.; Jockusch, B.M. Actin: From structural plasticity to functional diversity. *Eur. J. Cell Biol.* **2011**, *90*, 797–804. [CrossRef]
2. Rodrigues-Oliveira, T.; Wollweber, F.; Ponce-Toledo, R.I.; Xu, J.; Rittmann, S.K.M.; Klingl, A.; Pilhofer, M.; Schleper, C. Actin cytoskeleton and complex cell architecture in an Asgard archaeon. *Nature* **2023**, *613*, 332–339. [CrossRef] [PubMed]
3. Imachi, H.; Nobu, M.K.; Nakahara, N.; Morono, Y.; Ogawara, M.; Takaki, Y.; Takano, Y.; Uematsu, K.; Ikuta, T.; Ito, M.; et al. Isolation of an archaeon at the prokaryote–eukaryote interface. *Nature* **2020**, *577*, 519–525. [CrossRef] [PubMed]
4. Kovacs, J.; Song, J.; Auer, M.; He, J.; Hunter, W.; Wriggers, W. Correction of missing-wedge artifacts in filamentous tomograms by template-based constrained deconvolution. *J. Chem. Inf. Model.* **2020**, *60*, 2626–2633. [CrossRef]
5. Rogge, H.; Artelt, N.; Endlich, N.; Endlich, K. Automated segmentation and quantification of actin stress fibres undergoing experimentally induced changes. *J. Microsc.* **2017**, *268*, 129–140. [CrossRef]
6. Sazzed, S.; Song, J.; Kovacs, J.A.; Wriggers, W.; Auer, M.; He, J. Tracing actin filament bundles in three-dimensional electron tomography density maps of hair cell stereocilia. *Molecules* **2018**, *23*, 882. [CrossRef] [PubMed]
7. Sazzed, S.; Scheible, P.; He, J.; Wriggers, W. Spaghetti Tracer: A Framework for Tracing Semiregular Filamentous Densities in 3D Tomograms. *Biomolecules* **2022**, *12*, 1022. [CrossRef]
8. Rusu, M.; Starosolski, Z.; Wahle, M.; Rigort, A.; Wriggers, W. Automated tracing of filaments in 3D electron tomography reconstructions using Sculptor and Situs. *J. Struct. Biol.* **2012**, *178*, 121–128. [CrossRef]
9. Song, J.; Patterson, R.; Metlagel, Z.; Krey, J.F.; Hao, S.; Wang, L.; Ng, B.; Sazzed, S.; Kovacs, J.; Wriggers, W.; et al. A cryo-tomography-based volumetric model of the actin core of mouse vestibular hair cell stereocilia lacking plastin 1. *J. Struct. Biol.* **2020**, *210*, 107461. [CrossRef]
10. Rigort, A.; Günther, D.; Hegerl, R.; Baum, D.; Weber, B.; Prohaska, S.; Medalia, O.; Baumeister, W.; Hege, H.C. Automated segmentation of electron tomograms for a quantitative description of actin filament networks. *J. Struct. Biol.* **2012**, *177*, 135–144. [CrossRef]
11. Dimchev, G.; Amiri, B.; Fäßler, F.; Falcke, M.; Schur, F.K. Computational toolbox for ultrastructural quantitative analysis of filament networks in cryo-ET data. *J. Struct. Biol.* **2021**, *213*, 107808. [CrossRef] [PubMed]
12. Martins, B.; Sorrentino, S.; Chung, W.L.; Tatli, M.; Medalia, O.; Eibauer, M. Unveiling the polarity of actin filaments by cryo-electron tomography. *Structure* **2021**, *29*, 488–498. [CrossRef] [PubMed]
13. Schneider, J.; Jasnin, M. Capturing actin assemblies in cells using in situ cryo-electron tomography. *Eur. J. Cell Biol.* **2022**, *101*, 151224. [CrossRef] [PubMed]
14. Redemann, S.; Baumgart, J.; Lindow, N.; Shelley, M.; Nazockdast, E.; Kratz, A.; Prohaska, S.; Brugués, J.; Fürthauer, S.; Müller-Reichert, T. *C. elegans* chromosomes connect to centrosomes by anchoring into the spindle network. *Nat. Commun.* **2017**, *8*, 1–13. [CrossRef]
15. Loss, L.A.; Bebis, G.; Chang, H.; Auer, M.; Sarkar, P.; Parvin, B. Automatic segmentation and quantification of filamentous structures in electron tomography. In Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, Orlando, FL, USA, 7–10 October 2012; pp. 170–177.
16. Alioscha-Perez, M.; Benadiba, C.; Goossens, K.; Kasas, S.; Dietler, G.; Willaert, R.; Sahli, H. A robust actin filaments image analysis framework. *PLoS Comput. Biol.* **2016**, *12*, e1005063. [CrossRef]
17. Smith, M.B.; Li, H.; Shen, T.; Huang, X.; Yusuf, E.; Vavylonis, D. Segmentation and tracking of cytoskeletal filaments using open active contours. *Cytoskeleton* **2010**, *67*, 693–705. [CrossRef]
18. Chen, M.; Dai, W.; Sun, S.Y.; Jonasch, D.; He, C.Y.; Schmid, M.F.; Chiu, W.; Ludtke, S.J. Convolutional neural networks for automated annotation of cellular cryo-electron tomograms. *Nat. Methods* **2017**, *14*, 983–985. [CrossRef]
19. Image and Data Analysis Facility, Core Research Facilities of DZNE. Yapic. 2022. Available online: <https://yapic.github.io/yapic/> (accessed on 5 July 2023).
20. Tang, G.; Peng, L.; Baldwin, P.R.; Mann, D.S.; Jiang, W.; Rees, I.; Ludtke, S.J. EMAN2: An extensible image processing suite for electron microscopy. *J. Struct. Biol.* **2007**, *157*, 38–46. [CrossRef]

21. Mu, Y.; Sazzed, S.; Alshammari, M.; Sun, J.; He, J. A tool for segmentation of secondary structures in 3D cryo-EM density map components using deep convolutional neural networks. *Front. Bioinform.* **2021**, *1*, 51. [[CrossRef](#)]
22. Pettersen, E.F.; Goddard, T.D.; Huang, C.C.; Couch, G.S.; Greenblatt, D.M.; Meng, E.C.; Ferrin, T.E. UCSF Chimera—A visualization system for exploratory research and analysis. *J. Comput. Chem.* **2004**, *25*, 1605–1612. [[CrossRef](#)]
23. Fäßler, F.; Dimchev, G.; Hodirna, V.V.; Wan, W.; Schur, F.K. Cryo-electron tomography structure of Arp2/3 complex in cells reveals new insights into the branch junction. *Nat. Commun.* **2020**, *11*, 6437. [[CrossRef](#)] [[PubMed](#)]
24. Lawson, C.L.; Patwardhan, A.; Baker, M.L.; Hryc, C.; Garcia, E.S.; Hudson, B.P.; Lagerstedt, I.; Ludtke, S.J.; Pintilie, G.; Sala, R.; et al. EMDataBank unified data resource for 3DEM. *Nucleic Acids Res.* **2016**, *44*, D396–D403. [[CrossRef](#)] [[PubMed](#)]
25. Scheible, P.; Sazzed, S.; He, J.; Wriggers, W. TomoSim: Simulation of filamentous cryo-electron tomograms. In Proceedings of the 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Houston, TX, USA, 9–12 December 2021; pp. 2560–2565.
26. Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graph.* **1996**, *14*, 33–38. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.