

## Tracing Filaments in Simulated and Experimental 3D Cryo-Electron Tomography Maps Using a Fast Dynamic Programming Algorithm

Salim Sazzed<sup>1</sup>, Peter Scheible<sup>1</sup>, Jing He<sup>1</sup>, Manfred Auer<sup>2</sup> and Willy Wriggers<sup>3</sup>

<sup>1</sup>Dept. of Computer Science, Old Dominion University, Norfolk, VA 23529, United States, <sup>2</sup>Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States, <sup>3</sup>Dept. of Mechanical & Aerospace Engineering and Institute of Biomedical Engineering, Old Dominion University, Norfolk, VA 23529, United States

Cryo-electron tomography (cryo-ET) is an imaging technique used to assemble three-dimensional representations of biological macromolecules and cells. However, due to the limited electron dose that can be applied during imaging, they exhibit low signal-to-noise ratio (SNR) and “missing wedge” reconstruction artifacts caused by the effect of the limited sample tilt range on frequency domain amplitudes. Due to these inherent characteristics, automated segmentation of features from the cryo-ET image remains a challenging task. Our present work is mainly concerned with the efficient automated tracing of filamentous features, as applied to missing wedge corrected 3D maps [1] or outright tomograms [2,3]. We are also working on methods for computing simulated tomograms including noise and missing wedge artifacts for the validation of our algorithms. We present a novel fast dynamic programming algorithm for tracing actin filaments in 3D tomograms. The proposed algorithm can identify the high-density filament segments in any orientation by combining searches along the x, y, and z axes. Given an initial reference voxel of a filament (i.e., a seed point), it identifies the next reference voxel in each axis direction by tracing the highest density path for a user-defined step size  $s$  that can be calibrated to the nominal resolution of the tomogram or the separation of desired features. As in dynamic programming, the problem is solved optimally by breaking it into sub-problems (spatial decomposition into tracing windows defined by the step size  $s$ ) and then recursively finding the optimal solutions to the sub-problems within each tracing window. The tracing process shown in Figure 1 can deviate from the main axis directions (within a field of 45 degrees of each x, y and z) and it continues until it reaches the endpoint of the filament (decided by a threshold). We applied the proposed algorithm for tracing actin filaments in both experimental (Fig. 2A) and simulated cryo-ET maps (Fig 2B). The agreement with manual annotation [3] demonstrates the reliability of the proposed algorithm. Besides, the algorithm is computationally highly efficient because it traces each filament in a linear time, and it does not require a 6D rigid body search with template convolution [2]. Seed points are readily available in tracing applications to missing-wedge corrected maps [1], but it is harder to detect filaments in experimental tomograms [2,3]. We are currently adapting the method to make it fully automated without requiring user selected seed points and will present results at the conference. To further validate our tracing methods, our team is also working on generating simulated filament tomograms with the same noise and artifacts that would be common in experimental data. Using simulated data, we will have a known ground truth to evaluate our tracing results. This process gives us a quantitative approach to measuring our tracing technique’s accuracy. The goal is to create simulated data that realistically recreates the same obstacles presented in tracing an experimental map (Fig. 2B). To accomplish this, we start by creating the one-dimensional filament cores, either through procedural generation or by interpolating a manual trace. Volume is then given to the cores by convolution with a shape kernel. A noise model is then applied to the tomogram. We are working on analyzing the noise-models of experimental maps. For now, we are substituting simple Gaussian noise configured to create the desired signal-to-noise ratio. Finally, we simulate the missing-wedge artifact. The map is transformed to Fourier space using discrete FFT, where frequencies are masked according to the tilting range we wish to emulate.

$$(i_{new}, j_{new}, k_{new}) = \text{indices of} \left( \max \left\{ \begin{array}{l} \max(P_x(i_c + s, j_c + m, k_c + n), -s \leq m, n \leq +s) \\ \max(P_y(i_c + m, j_c + s, k_c + n), -s \leq m, n \leq +s) \\ \max(P_z(i_c + m, j_c + n, k_c + s), -s \leq m, n \leq +s) \end{array} \right\} \right)$$

$$P_x(i, j, k) = D(i, j, k) + \max(P_x(i - 1, j + m, k + n), m, n \in \{-1, 0, 1\})$$

$$P_y(i, j, k) = D(i, j, k) + \max(P_y(i + m, j - 1, k + n), m, n \in \{-1, 0, 1\})$$

$$P_z(i, j, k) = D(i, j, k) + \max(P_z(i + m, j + n, k - 1), m, n \in \{-1, 0, 1\})$$

Figure 1. Figure 1. The equation is used to identify the next reference voxel ( $i_{new}$ ,  $j_{new}$ ,  $k_{new}$ ) recursively within each tracing window.  $D$  denotes the voxel density,  $(i_c, j_c, k_c)$  are the current reference voxel indices, and the  $P_x/y/z$  represent the local path densities from the reference voxel along the main axis directions.

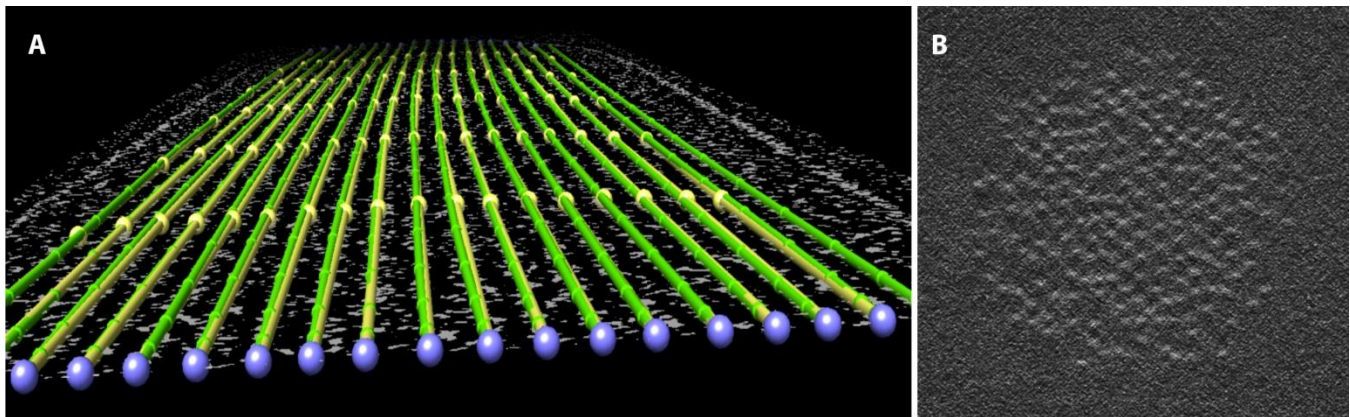


Figure 2. Figure 2. (A) Comparison between manually and computationally traced filaments (yellow and green, respectively; seed points are blue) on a z section of mouse vestibular hair cell stereocilia tomogram [3]. (B) y section of a simulated tomogram generated from the full manual tracing in [3].

#### References

- [1] J Kovacs *et al*, J. Chem. Inf. Model., 60 (2020) p. 2626.
- [2] M Rusuet *et al*, J. Struct. Biol., 178 (2012) p. 121.
- [3] J Song *et al*, J. Struct. Biol., 210 (2020) p. 107461.
- [4] The work in this paper was supported in part by NIH R01-GM062968.