

Guide frames improve centerline estimation of complex postures in WormTracer

Chung-Kuan Chen¹, Yuichi Iino¹, Yu Toyoshima^{1,2§}

¹Department of Biological Sciences, Graduate School of Science, The University of Tokyo, Tokyo, 113-0022, Japan

²Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Chiba, 277-8562, Japan

[§]To whom correspondence should be addressed: ytoyo@bs.s.u-tokyo.ac.jp

Abstract

Precise estimation of body posture is essential for analyzing animal behavior. *C. elegans* exhibits a variety of complex postures, such as turning and looping, for which centerline extraction is challenging. WormTracer, an algorithm designed to ensure temporal continuity of centerlines, enabled reliable centerline extraction. However, accurate estimation remains difficult without human intervention in animals exhibiting prolonged, complex postures. Here, we present an updated version of WormTracer that accepts user-revised centerlines as guide frames, enabling accurate estimation even during prolonged complex postures. In addition, we provide a Napari plugin, Napari-WormTracer, which allows users to manually revise centerlines estimated by WormTracer.

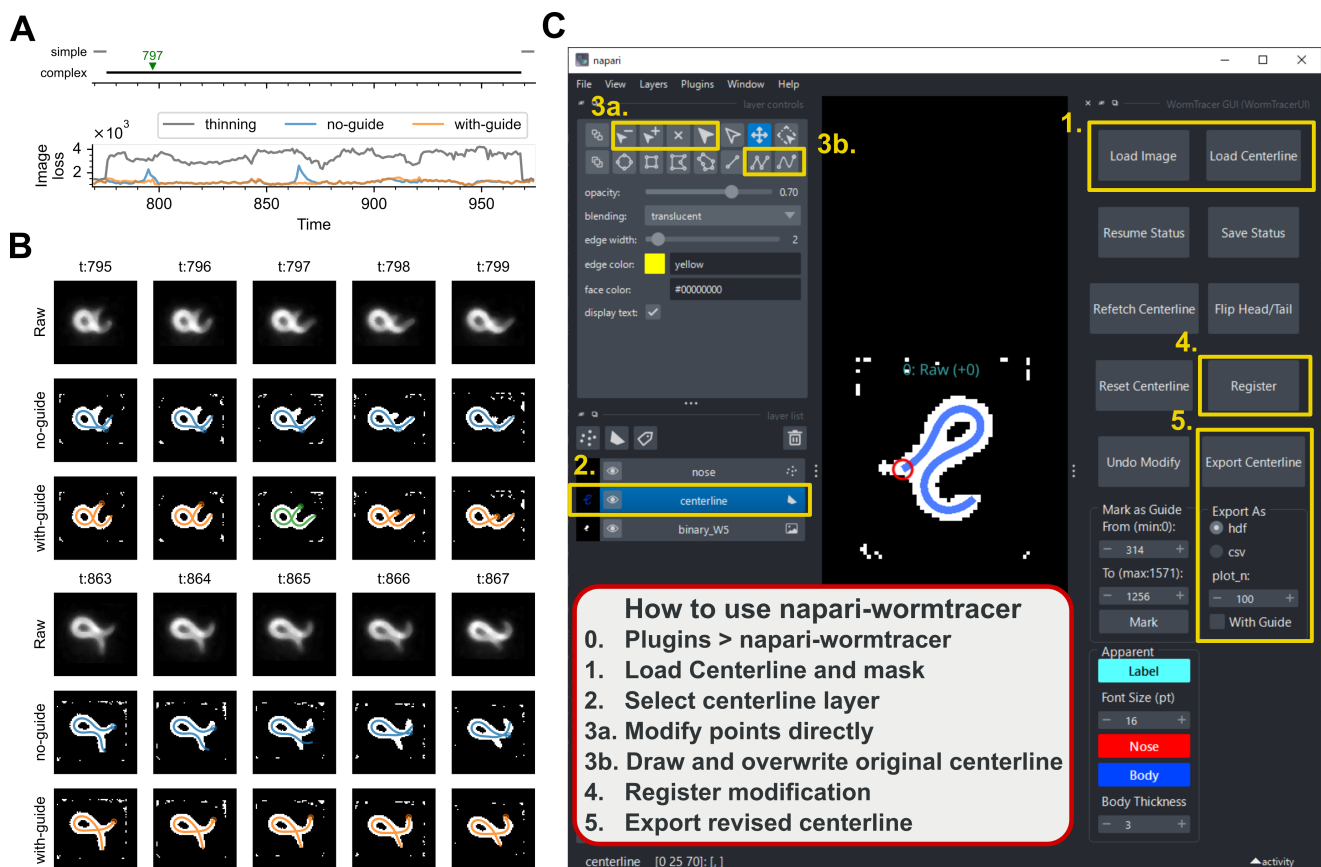


Figure 1. Improved WormTracer enables accurate estimation of complex posture using a revised guide frame:

(A) Time-series analysis of a representative worm exhibiting prolonged complex postures. The first panel shows posture classification (simple vs. complex), with guide frames indicated in green. The second panel shows image loss (MSE) for centerlines obtained using the canonical thinning algorithm (gray) and after WormTracer optimization either without (blue) or with (orange) guide frames. (B) Time-series raw images and corresponding binarized mask overlaid with the centerlines estimated without or with guide frame at time point 797. Circle indicates head position. Centerlines obtained without guide frames are shown in blue, whereas those obtained with guide frames are shown in orange. A manually annotated centerline used as a guide frame is shown in green. (C) Interface of Napari-Wormtracer, a Napari-based plugin for revising centerlines exported from WormTracer, is shown.

Description

Behavioral analysis relies on accurate extraction of body posture, which is essential for uncovering the mechanisms underlying motion control and for quantitatively modeling the movement. However, precise posture estimation remains challenging due to the diversity of postures, frequent occlusion of body parts, and variability in imaging conditions (Mathis et al., 2018; Pereira et al., 2022). In *C. elegans* research, many studies have been dedicated to estimating the body posture using variety of computational approaches. For example, Tierpsy provides a well-established pipeline to analyze the worm posture using traditional imaging processing techniques (Javer, Currie, et al., 2018; Javer, Ripoll-Sánchez, et al., 2018). Moreover, deep learning models for instance segmentation have improved posture estimation by increasing the accuracy of binary masks (Castro et al., 2025; Deserno & Bozek, 2023). In addition to the mask-based extraction, several methods directly predict the centerlines from raw input images without prior binarization (Alonso & Kirkegaard, 2023; Hebert et al., 2021; Layana Castro et al., 2023; Saha et al., 2025; Weheliye et al., 2025). However, training such deep learning model typically requires large amounts of manually annotated data or generation of synthetic data from limited annotations. To address this limitation, in our previous study we developed WormTracer (Kuze et al., 2026), an algorithm that estimates the centerline by comparing a worm-shaped mask reconstructed from the candidate centerline with the observed worm silhouette, while enforcing temporal continuity and spatial smoothness of the centerlines. First, WormTracer preprocessed the input masks to extract the centerlines using a canonical thinning algorithm. The extracted centerlines were then classified as either “simple” or “complex”. For example, during turning behavior, contact between the head or tail and the body generated looped masks that could not be accurately resolved using the thinning algorithm alone. Consequently, these frames were classified as “complex”. To objectively identify the complex frames based on the estimated centerlines, WormTracer evaluated the pixel mismatch between the original binary body mask and the mask reconstructed from the centerlines. Frames in which the thinning algorithm failed to accurately reconstruct the original mask were classified as complex (Fig. 1A). Consecutive frames with the same classification were then grouped into behavioral sequences. Following this preprocessing step, initial centerlines for the simple sequences were derived directly from the thinning method, whereas those for the complex sequences were generated by linearly interpolating centerlines from adjacent simple sequences. These estimates were then optimized via gradient descent to minimize mask pixel mismatch, with a regularization penalty for abrupt spatial and temporal changes. Eventually, the output centerlines were optimized to best fit the input mask while maintaining their spatial and temporal continuity. The entire optimization required only a time-series binary mask, and the centerlines were automatically estimated without human intervention.

Despite these advantages, WormTracer has several limitations. When complex postures persist for a long period, relying on adjacent simple frames for initial estimates becomes insufficient. The errors are more likely to occur in highly coiled shapes, where mask-based reconstruction alone may fail to resolve the correct centerline topology. An example dataset shown here exhibited prolonged complex posture from time points 776 to 968 and was analyzed using WormTracer to estimate centerlines (Fig. 1A, blue). In this case, frames around time point 797 showed increased image loss. Consistently, these frames also displayed incorrect centerline predictions (Fig. 1B, blue).

To address these limitations, we updated WormTracer to incorporate manually specified centerlines (termed guide frames) for optimization. The guide frames serve as anchoring points that are treated as ground truth during optimization and partitions the full sequence into smaller optimization blocks. During optimization, these guide frames correct the topology of the centerlines, allowing the continuity constraints of the WormTracer algorithm to propagate the correct head-tail orientation and bending angles to neighboring frames. As a result, providing a single guide frame at time point 797 (Fig. 1A and B, green) corrected the centerline topology not only at that frame but also during the complex postures around time point 865 within the same behavioral sequence (Fig. 1A and B, orange). This result indicates that guide frames can resolve erroneous centerline topology and allow WormTracer to automatically propagate the correction throughout the sequence, substantially reducing the amount of manual intervention required.

In conclusion, incorporating guide frames improves overall prediction accuracy and enables WormTracer to recover correct centerlines with minimal manual intervention. The updated version of WormTracer is available at: <https://github.com/lycantrope/wormtracer/releases/tag/v21.0.0>

In addition to enabling WormTracer to incorporate guide frames for centerline refinement, we developed a Napari plugin (Sofroniew et al., 2026), Napari-WormTracer (Fig. 1C; <https://github.com/lycantrope/napari-wormtracer>). This tool enables users to easily inspect centerlines generated by WormTracer. It also provides functionality to manually revise centerlines and register these corrections, facilitating the generation of guide frame data for subsequent WormTracer optimization. Furthermore, it can serve as a practical tool to prepare training datasets for other established methods.

In summary, guide frames enhance the robustness of WormTracer for complex postures, and Napari-WormTracer provides a practical interface for manual correction, enabling accurate centerline estimation with minimal user intervention.

Methods

Strains and culture

All *C. elegans* strains were mainly cultured on nematode growth medium (NGM) with *Escherichia coli* strain OP50, as described (Brenner, 1974) and maintained at 20°C. For sample movies, bright field images were acquired from adult individuals of TOY21, a strain derived from QD15#4 (Toyoshima et al., 2020; Yang et al., 2026) and was developed for whole-brain calcium imaging.

Definition of loss functions for centerline estimation

WormTracer defines five loss functions to evaluate the fitness of estimated centerlines to the observed worm mask, which ensures both spatial accuracy and temporal continuity (Kuze et al., 2026). These include image loss, temporal angular continuity loss (continuity loss), spatial angular continuity loss (smoothing loss), length continuity loss (length loss), and centroid loss (center loss). Specifically, image loss is calculated as the mean squared error (MSE) between two masks, the observed worm silhouette and the mask reconstructed from the estimated centerline. Temporal and spatial angular continuity losses are defined as the MSE differences of segment bending angles along the temporal dimension (i.e., between consecutive time points) and the spatial dimension (i.e., along the centerline within a single frame), respectively, constraining abrupt changes in angle. Length continuity loss and centroid loss enforce consistency in segment length and stability of the center of mass by minimizing their variance and displacement, respectively.

Implementation of WormTracer with guide frames

The major difference from the previous version of WormTracer was the support for incorporating guide frames. Particularly, the newer of WormTracer loaded the correct centerlines from external files. After calculating the image loss as, $Im_{loss,t} = MSE(mask_{model,t} - mask_{real,t})$, the losses in the frames with correct centerlines (guide frames) were replaced by the minimum loss, expressed as $Im_{loss,t_g} = min(Im_{loss,t}, t_g \in guide_frames)$. Subsequently, the long behavioral sequences were split into small optimization blocks at these guide frames, thereby anchoring the topology of neighboring centerlines, such as head-tail orientation, segment length and bending angles.

WormTracer was implemented in Python using several libraries for data analysis and image processing, including NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), scikit-image (Van Der Walt et al., 2014), and OpenCV (Bradski, 2000). Deep learning components were implemented using PyTorch (Paszke et al., 2019). Visualization was performed using Matplotlib (Hunter, 2007), and the final figure layouts were assembled using Inkscape (<https://inkscape.org/>). All code and detailed usage instructions are available at <https://github.com/lycantrope/wormtracer> and <https://github.com/lycantrope/napari-wormtracer>. In this paper, we used WormTracer v21.0.0 and napari-wormtracer 2026.05.29.

Reagents

Strain	Genotype	Source
OP50	<i>Escherichia coli</i>	CGC
TOY21	<i>qjIs11[glr-1p::svnls2::TagBFPsyn,ser-2(prom2)p::svnls2::TagBFPsyn] V; peIs3042[eat-4p::svnls2::TagRFP675syn,lin-44p::GFP] X.; Is[H20p::nls3::tdTomato,H20p::nls2::GCaMP6f]#72-2-5</i>	(Yang et al., 2026)

Acknowledgements: We are grateful to Jingxuan Zeng for providing the sample videos and handwritten centerlines used in this work.

Extended Data

Description: A python package for centerlines estimation. Resource Type: Software. File: [WormTracer-21.0.0.zip](#). DOI: [10.22002/11rsh-36g19](#)

Description: A napari-plugin for centerlines refinement. Resource Type: Software. File: [napari-wormtracer-2026.05.29.zip](#). DOI: [10.22002/f4mjs-j0x86](#)

References

Alonso A, Kirkegaard JB. 2023. Fast detection of slender bodies in high density microscopy data. Communications Biology. 6: 754. DOI: [10.1038/s42003-023-05098-1](#)

Brenner S. 1974. THE GENETICS OF CAENORHABDITIS ELEGANS. Genetics. 77: 71. DOI: [10.1093/genetics/77.1.71](#)

- Castro PEL, Kounakis K, Garvi AG, Gkikas I, Tsiamantas I, Tavernarakis N, Sanchez Salmeron AJ. 2025. SegElegans: Instance segmentation using dual convolutional recurrent neural network decoder in *Caenorhabditis elegans* microscopic images. *Computers in Biology and Medicine*. 190: 110012. DOI: [10.1016/j.combiomed.2025.110012](https://doi.org/10.1016/j.combiomed.2025.110012)
- Deserno M, Bozek K. 2023. WormSwin: Instance segmentation of *C. elegans* using vision transformer. *Scientific Reports*. 13: 11021. DOI: [10.1038/s41598-023-38213-7](https://doi.org/10.1038/s41598-023-38213-7)
- Harris CR, Millman KJ, Van Der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al., Oliphant TE. 2020. Array programming with NumPy. *Nature*. 585: 357. DOI: [10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2)
- Hebert L, Ahamed T, Costa AC, O Shaughnessy L, Stephens GJ. 2021. WormPose: Image synthesis and convolutional networks for pose estimation in *C. elegans*. *PLOS Computational Biology*. 17: e1008914. DOI: [10.1371/journal.pcbi.1008914](https://doi.org/10.1371/journal.pcbi.1008914)
- Hunter JD. 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 9: 90. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55)
- Javer A, Currie M, Lee CW, Hokanson J, Li K, Martineau CN, et al., Brown AEX. 2018. An open-source platform for analyzing and sharing worm-behavior data. *Nature Methods*. 15: 645. DOI: [10.1038/s41592-018-0112-1](https://doi.org/10.1038/s41592-018-0112-1)
- Javer A, Ripoll Sanchez L, Brown AEX. 2018. Powerful and interpretable behavioural features for quantitative phenotyping of *Caenorhabditis elegans*. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 373: 20170375. DOI: [10.1098/rstb.2017.0375](https://doi.org/10.1098/rstb.2017.0375)
- Kuze K, Tazawa UT, Suwazono K, Chen CK, Toyoshima Y, Iino Y. 2026. WormTracer: A precise method for worm posture analysis using temporal continuity. *Journal of Neuroscience Methods*. 427: 110644. DOI: [10.1016/j.jneumeth.2025.110644](https://doi.org/10.1016/j.jneumeth.2025.110644)
- Layana Castro PE, Garcia Garvi A, Navarro Moya F, Sanchez Salmeron AJ. 2023. Skeletonizing *Caenorhabditis elegans* Based on U-Net Architectures Trained with a Multi-worm Low-Resolution Synthetic Dataset. *International Journal of Computer Vision*. 131: 2408. DOI: [10.1007/s11263-023-01818-6](https://doi.org/10.1007/s11263-023-01818-6)
- Mathis A, Mamidanna P, Cury KM, Abe T, Murthy VN, Mathis MW, Bethge M. 2018. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*. 21: 1281. DOI: [10.1038/s41593-018-0209-y](https://doi.org/10.1038/s41593-018-0209-y)
- Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al., Chintala S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. DOI: [10.48550/ARXIV.1912.01703](https://doi.org/10.48550/ARXIV.1912.01703)
- Pereira TD, Tabris N, Matsliah A, Turner DM, Li J, Ravindranath S, et al., Murthy M. 2022. SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods*. 19: 486. DOI: [10.1038/s41592-022-01426-1](https://doi.org/10.1038/s41592-022-01426-1)
- Saha D, Chaudhary S, Vyas D, Ghosh Roy A, Sharma R. 2025. Deep-Pose-Tracker: an automated behavioural analysis framework for *Caenorhabditis elegans*. DOI: [10.1101/2025.11.23.689997](https://doi.org/10.1101/2025.11.23.689997)
- Sofroniew N, Lambert T, Bokota G, Nunez Iglesias J, Sobolewski P, Sweet A, et al., Yadav AS. 2026. napari: a multi-dimensional image viewer for Python. DOI: [10.5281/ZENODO.3555620](https://doi.org/10.5281/ZENODO.3555620)
- Toyoshima Y, Wu S, Kanamori M, Sato H, Jang MS, Oe S, et al., Iino Y. 2020. Neuron ID dataset facilitates neuronal annotation for whole-brain activity imaging of *C. elegans*. *BMC Biology*. 18: 30. DOI: [10.1186/s12915-020-0745-2](https://doi.org/10.1186/s12915-020-0745-2)
- Van Der Walt S, Schonberger JL, Nunez Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, Yu T. 2014. scikit-image: image processing in Python. *PeerJ*. 2: e453. DOI: [10.7717/peerj.453](https://doi.org/10.7717/peerj.453)
- Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, et al., Vazquez Baeza Y. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*. 17: 261. DOI: [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2)
- Weheliye WH, Rodriguez J, Feriani L, Javer A, Uhlmann V, Brown AEX. 2025. A neural network model enables worm tracking in challenging conditions and increases signal-to-noise ratio in phenotypic screens. *PLOS Computational Biology*. 21: e1013345. DOI: [10.1371/journal.pcbi.1013345](https://doi.org/10.1371/journal.pcbi.1013345)
- Yang X, Zeng J, Chen CK, Iino Y, Toyoshima Y. 2026. Developing a Robust Multiround HCR - FISH Method Modified for *Caenorhabditis elegans*. *Genes to Cells*. 31: e70098. DOI: [10.1111/gtc.70098](https://doi.org/10.1111/gtc.70098)

Funding: This work was supported by the Japan Science and Technology Agency (JST) CREST program, "Establishment of high-speed, high-dimensional closed-loop optical measurement technology and its applications to neuroscience" (JPMJCR22N4), as well as its satellite AIP challenge program 2025 granted to C.-K. C.

Conflicts of Interest: The authors declare that there are no conflicts of interest present.

Author Contributions: Chung-Kuan Chen: software, writing - original draft, writing - review editing, visualization. Yuichi Iino: conceptualization, writing - review editing, software. Yu Toyoshima: conceptualization, funding acquisition,

8/31/2026 - *Open Access*

project administration, writing - review editing.

Reviewed By: Anonymous

WormBase Paper ID: WBPaper00070116

History: **Received** June 7, 2026 **Revision Received** August 15, 2026 **Accepted** August 27, 2026 **Published Online** August 31, 2026 **Indexed** September 14, 2026

Copyright: © 2026 by the authors. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International (CC BY 4.0) License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Citation: Chen CK, Iino Y, Toyoshima Y. 2026. Guide frames improve centerline estimation of complex postures in WormTracer. microPublication Biology. [10.17912/micropub.biology.002235](https://doi.org/10.17912/micropub.biology.002235)