TrackDLO: Tracking Deformable Linear Objects Under Occlusion with Motion Coherence

Jingyi Xiang[1], Holly Dinkel[2], Harry Zhao[3], Naixiang Gao[1], Brian Coltin[4], Trey Smith[4], Timothy Bretl[2]

***Abstract*— The TrackDLO algorithm estimates the shape of a Deformable Linear Object (DLO) under occlusion from a sequence of RGB-D images. TrackDLO is vision-only and runs in real-time. It requires no external state information from physics modeling, simulation, visual markers, or contact as input. The algorithm improves on previous approaches by addressing three common scenarios which cause tracking failures: tip occlusion, mid-section occlusion, and self-occlusion. This is achieved through the application of Motion Coherence Theory to impute the spatial velocity of occluded nodes, the use of the topological geodesic distance to track self-occluding DLOs, and the introduction of a non-Gaussian kernel that only penalizes lower-order spatial displacement derivatives to reflect DLO physics. Improved real-time DLO tracking under mid-section occlusion, tip occlusion, and self-occlusion is demonstrated experimentally. The source code and demonstration data are publicly released.**

I. Introduction

This work presents TrackDLO [1], an algorithm for real-time tracking of Deformable Linear Object (DLO) (e.g., rope, wire, tubing) shapes. The TrackDLO algorithm tracks DLOs in RGB-D imagery for manipulation tasks, such as knot tying or wire routing, or to monitor DLOs for collision prevention [2]–[7]. These tasks are common in applications including robotic surgery, industrial automation, power line avoidance, and human habitat maintenance [8]–[12]. Several methods track DLOs in real-time with and without visual markers and physics simulation [13]–[19]. The TrackDLO algorithm builds on existing tracking work by addressing three scenarios common in manipulation tasks which cause tracking failures: tip occlusion, mid-section occlusion, and self-occlusion. This work makes the following listed contributions:

[1]Jingyi Xiang and Naixiang Gao are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. e-mail: {jingyix4, ngao4}@illinois.edu

[2]Holly Dinkel and Timothy Bretl are with the Department of Aerospace Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. e-mail: {hdinkel2, tbretl}@illinois.edu

[3]Harry Zhao is with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305 USA. e-mail: harzhao@stanford.edu

[4]Brian Coltin and Trey Smith are with the Intelligent Robotics Group, NASA Ames Research Center, Moffett Field, CA, 94035 USA. e-mail: {brian.coltin, trey.smith}@nasa.gov.

This work was previously published as [1] and authored with J. Xiang, H. Zhao, N. Gao, B. Coltin, T. Smith, and T. Bretl.; ©2023 IEEE. Reprinted, with permission, from J. Xiang, H. Dinkel, H. Zhao, N. Gao, B. Coltin, T. Smith, and T. Bretl, "TrackDLO: Tracking Deformable Linear Objects Under Occlusion with Motion Coherence," in *IEEE Robot. Automat. Lett.*, vol. 8, no. 10, pp. 6179-6186, Oct. 2023, doi:10.1109/LRA.2023.3303710.

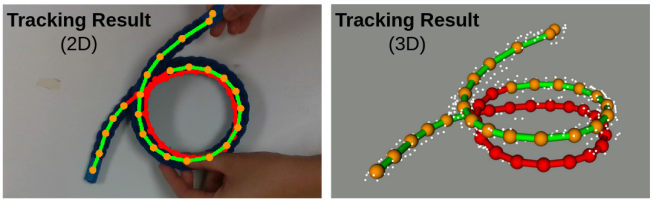

Fig. 1. TrackDLO performs occlusion-robust 3D DLO tracking without physics simulation or other external state information using motion coherence. TrackDLO accurately predicts the location for occluded nodes (red) by imputing their spatial velocities from visible nodes (orange) under mid-section occlusion, tip occlusion, and self-occlusion.

1) TrackDLO accurately tracks DLOs in real-time with tip occlusion without information from modeling, simulation, visual markers, or contact. This is achieved by preserving the total length of the DLO and using Motion Coherence Theory (MCT) to impute the spatial velocity of occluded tip nodes from the spatial velocity of visible nodes [20]. The spatial velocity is used to update the DLO shape estimate.
2) TrackDLO tracks self-occluding DLOs without modeling entanglement. This is achieved by incorporating the geodesic distance into the kernel describing how pairs of nodes influence each other's motion.
3) TrackDLO reflects the physics of DLOs by introducing a kernel which only penalizes lower-order spatial velocity derivatives. This work analytically derives forms the kernel can take to satisfy optimality in the Expectation-Maximization algorithm.
4) Data and source code for DLO tracking are released at: https://github.com/RMDLO/trackdlo. The supplementary video demonstrates tracking performance on two DLOs with different material properties and is shared at: https://youtu.be/MxqNJsen5eg.

II. Related Work

The Coherent Point Drift (CPD) algorithm performs non-rigid registration to map one set of points onto another. The CPD algorithm uses Gaussian Mixture Model (GMM) clustering and Motion Coherence Theory (MCT) with Expectation-Maximization (EM) to find the probability distribution parameters which maximize the likelihood that a predicted point set corresponds to the original point set [20]–[23]. The Global-Local Topology Preservation (GLTP) algorithm performs modified non-rigid point set registration. The objective function for EM in GLTP uses CPD with locally linear embedding to preserve local topology [24].

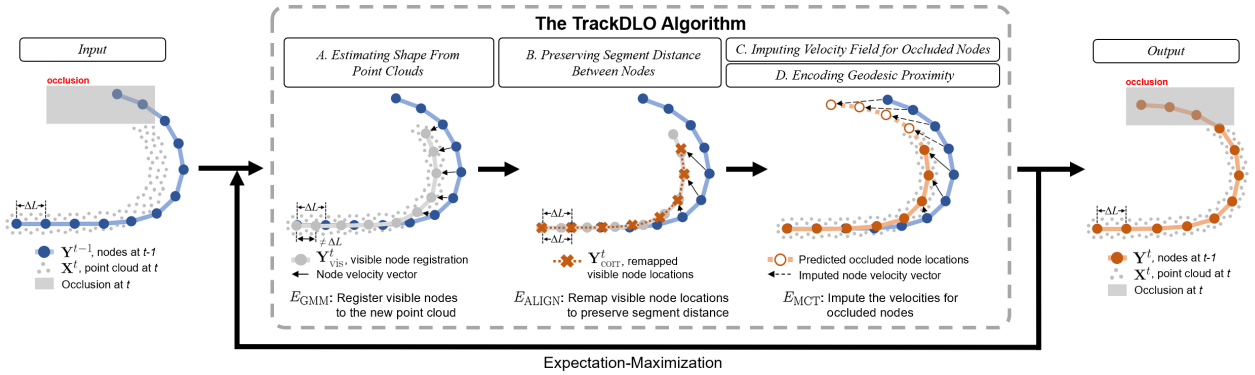

Fig. 2. The TrackDLO algorithm registers visible nodes to a point cloud, preserves segment distance between nodes, imputes the velocity field for occluded nodes, and encodes geometric proximity to accurately track a DLO under occlusion.

Non-rigid point set registration from CPD and GLTP is the foundation for the Constrained Deformable CPD (CDCPD) algorithm which also constrains DLO stretching and recovers from tracking failures [16]. The CDCPD2 algorithm builds on CDCPD by incorporating known correspondences, self-intersection constraints, and obstacle interaction constraints to accurately track under tip and large-scale occlusion [17]. More recent work performs DLO tracking under occlusion with particle filtering on a lower-dimensional latent space embedding learned with an autoencoder [25] and learning-based shape estimation under occlusion [26]. These existing methods require additional information from modeling, simulation, visual markers, or contact for accuracy or show inconsistent performance under occlusion.

III. The TrackDLO Algorithm

The TrackDLO algorithm is shown in Figure 2. For N *points* in 3D received at time t from a depth sensor, $\mathbf{X}^t_{N\times 3} = (\mathbf{x}^t_1, \ldots, \mathbf{x}^t_N)^T$, TrackDLO represents the DLO shape with a collection of M ordered *nodes*, $\mathbf{Y}^t_{M\times 3} = (\mathbf{y}^t_1, \ldots, \mathbf{y}^t_M)^T$, and the edges connecting adjacent nodes. The measurement $\mathbf{X}^t$ can contain outliers due to noise and it can be incomplete due to occlusion. The TrackDLO algorithm is initialized by de-projecting a pixel chain output by an occlusion-robust DLO detection algorithm into 3D and sampling nodes equally distributed along the chain [27]. The CPD objective function which solves for $\mathbf{Y}^t$ through EM is first modified by performing Gaussian Mixture Model (GMM) clustering with modified membership probability [16]. Next, visible node locations are estimated and remapped to preserve the DLO segment lengths and total length. The velocity field for occluded nodes is imputed using Motion Coherence Theory (MCT) [20], and the geodesic proximity is used to accurately represent distances between nodes to enable tracking in the presence of self-occlusion.

IV. Experiments

In experiments analyzing the accuracy of TrackDLO, CDCPD2 with and without optional gripper information, CDCPD, and GLTP, TrackDLO achieves the lowest node-to-PWL curve frame error and second-fastest computation time. All experiments used an Intel RealSense d435 camera with $k_{\text{vis}} = 500, \beta = 0.5, \lambda = 5 \times 10^4$, $\alpha = 3$, $w = 20$ pixels, $\tau_{\text{vis}} = 5\text{mm}$, $d_{\text{vis}} = 4\text{cm}$, and convergence of EM when the average node position change between iterations is less than 0.2mm.

Experiments comparing the performance of TrackDLO, CDCPD2 with and without optional gripper information, CDCPD, and GLTP under occlusion used the frame error metric defined as follows. The set of points $\text{PWL}(\mathbf{Y})$ in the PWL curve representation of an ordered node sequence $\mathbf{Y}$ includes both the nodes themselves and all points in the line segments connecting nodes. The point-set distance is

$$d(\mathbf{y}, \mathcal{S}) = \inf_{\mathbf{y}' \in \mathcal{S}} \|\mathbf{y} - \mathbf{y}'\|, \tag{1}$$

where $\mathbf{y}$ is a point and $\mathcal{S}$ is a set of points. The per-node error between two node sequences is

$$\varepsilon(\mathbf{Y}^t, \mathbf{Y}^t_{\text{true}}) = \frac{1}{M} \sum_{\mathbf{y}^t_i \in \mathbf{Y}^t} d(\mathbf{y}^t_i, \text{PWL}(\mathbf{Y}^t_{\text{true}})), \tag{2}$$

and the frame error metric, mirroring the per-node error to guarantee symmetry, is

$$\mathcal{E}(\mathbf{Y}^t_{\text{true}}, \mathbf{Y}^t) = \frac{1}{2}\left(\varepsilon(\mathbf{Y}^t_{\text{true}}, \mathbf{Y}^t) + \varepsilon(\mathbf{Y}^t, \mathbf{Y}^t_{\text{true}})\right). \tag{3}$$

Evaluation was performed for three scenarios:

1) *Stationary*–This scenario tests tracking error accumulation and length preservation under tip occlusion.
2) *Perpendicular Motion*–This scenario tests DLO tracking accuracy when both tips are visible and the mid-section is occluded.
3) *Parallel Motion*–This scenario tests DLO length preservation and tracking accuracy when one tip moves through occlusion.

For each scenario, RGB-D image and point data were saved in a Robot Operating System (ROS) bag file. Occlusion was injected by removing pixels within a bounding box area in the DLO segmentation mask. The blue rope DLO was evenly marked with red tape which was segmented by thresholding on the red and blue colors. In each of the red and blue segmentation masks, contour filtering and blob detection identified the blue and red segments along the DLO and keypoint detection returned their centroids. These centroids were combined to form the ground truth nodes, $\mathbf{Y}^t_{\text{true}}$, which

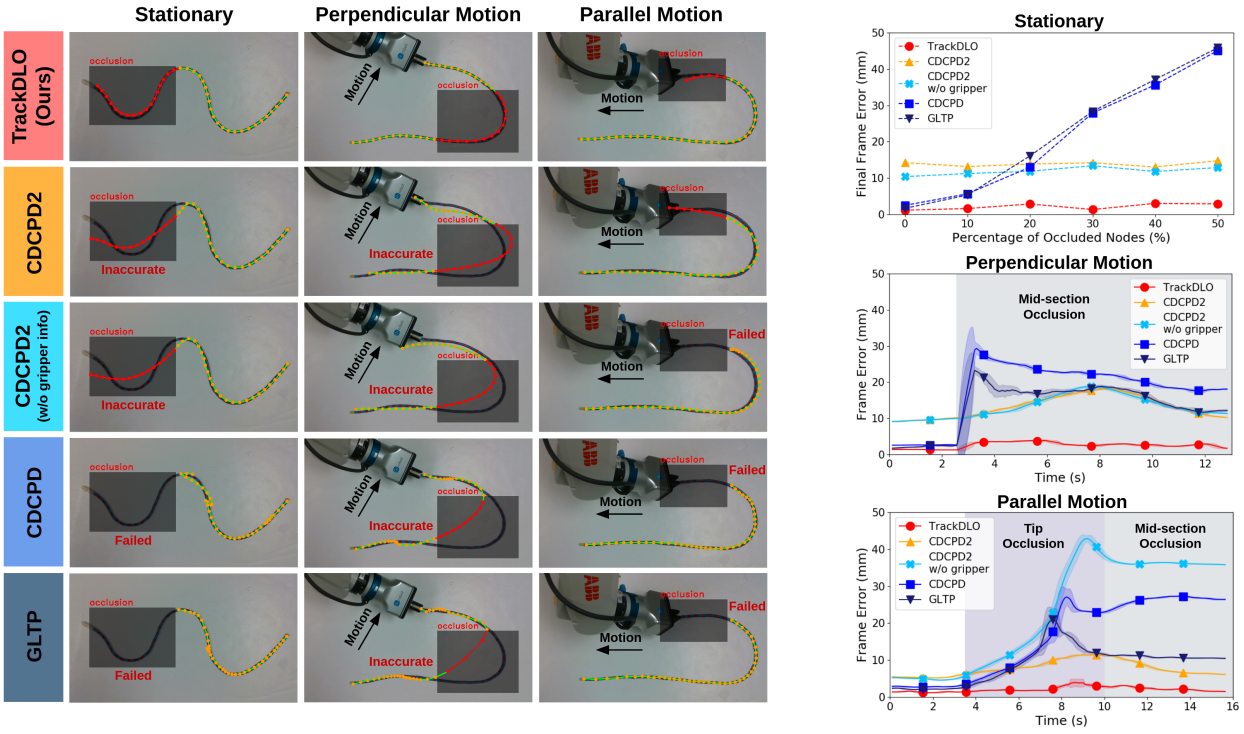

Fig. 3. Compared to CDCPD2 with and without gripper information, CDCPD, and GLTP, TrackDLO accurately estimates the state of the DLO under scaled, tip, and mid-section occlusion. Among these algorithms, TrackDLO had the lowest frame error.

were compared to the tracked nodes in $\mathbf{Y}^t$. Note the markers on the DLO were only used for evaluation and were not required by any algorithm for tracking. Each experiment was repeated 10 times for each algorithm in each scenario.

Evaluation results are reported in Figure 3. In the *Stationary* scenario, the occlusion size is scaled as a percentage of the number of nodes along the DLO and the DLO is occluded for 25 seconds after initialization. The TrackDLO algorithm achieves the lowest average final frame error as a function of percentage of occlusion for the *Stationary* scenario. In the *Perpendicular Motion* scenario, occlusion is injected in the mid-section of the DLO three seconds after tracking begins before the object moves and remains there until the end of the bag file. For the CDCPD2, CDCPD, and GLTP algorithms, the error increases after the injection of occlusion and decreases as the tracking estimates begin to catch up with the DLO state. The TrackDLO algorithm achieves the lowest average frame error in this scenario. In the *Parallel Motion* scenario, the DLO is initially fully visible and the gripper moves the DLO tip through an occluded region. Tracking error increases for all algorithms until the DLO tip becomes visible again during the mid-section occlusion period. The TrackDLO algorithm achieves the lowest frame error in this scenario as well.

Timing experiments were conducted to compare the speeds of the TrackDLO, CDCPD2, CDCPD, and GLTP algorithms. The times reported in Table I were obtained on a computer with a Ryzen Threadripper 3960X CPU and 64 GB RAM. The CDCPD algorithm is implemented in Python while TrackDLO, CDCPD2, and GLTP are implemented in C++ using original repositories where available. Times include computation time for each algorithm after receiving data. Times exclude interfacing with ROS, segmenting object masks, and downsampling point clouds. Each algorithm was run on the *Perpendicular Motion* and *Parallel Motion* ROS bag files for 10 trials. Times are the average computation times across each algorithm run on each frame of the 10 trials of the two motion scenarios. The TrackDLO algorithm achieves the second-fastest average computation time.

TABLE I
TRACKING TIMING COMPARISON (IN MS)

Algorithm	**TrackDLO (Ours)**	CDCPD2	CDCPD	GLTP
Tracking Step	9.20	11.42	18.14	**5.43**

V. CONCLUSIONS AND FUTURE WORK

This work introduced TrackDLO, a real-time, vision-only algorithm for occluded DLO tracking. TrackDLO is robust under three types of occlusion and produced the lowest tracking error when compared to the CDCPD2, CDCPD, and GLTP algorithms. Source code, data, and a video of examples of DLO tracking are released. The limitations of TrackDLO include the requirements of good depth resolution, small motion between frames, and motion of occluded nodes to be reflected in the motion of visible nodes. Future work could track multiple DLOs as they move in cluttered environments [28]–[32], track a DLO with multiple cameras, track a DLO as multiple sections of it are occluded (where the occlusion size is significantly greater than d_{vis}), integrate DLO tracking into closed-loop DLO shape control [6], [33], integrate DLO tracking into knot planning [34], or provide a broader comparison to methods based on learning (e.g., [25] and [26]) rather than on point set registration.

ACKNOWLEDGEMENTS

The authors thank Dylan Colli for assistance with the CDCPD2 repository, Paulo V.K. Borges for providing feedback on this manuscript, the members of the Representing and Manipulating Deformable Linear Objects project for their support, and the teams developing and maintaining the open-source software used in this project [35]–[42]. Jingyi Xiang performed part of this work with the Illinois Space Grant Consortium Undergraduate Research Opportunity Program, and NASA Space Technology Graduate Research Opportunity award 80NSSC21K1292 supported Holly Dinkel.

REFERENCES

[1] J. Xiang, H. Dinkel, H. Zhao, N. Gao, B. Coltin, T. Smith, and T. Bretl, "TrackDLO: Tracking Deformable Linear Objects Under Occlusion With Motion Coherence," *IEEE Robot. Autom. Lett.*, vol. 8, no. 10, pp. 6179–6186, 2023.

[2] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects," vol. 5, no. 2, 2020, pp. 2372–2379.

[3] M. Yan, G. Li, Y. Zhu, and J. Bohg, "Learning Topological Motion Primitives for Knot Planning," in *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, 2020.

[4] R. Lagneau, A. Krupa, and M. Marchal, "Automatic Shape Control of Deformable Wires Based on Model-Free Visual Servoing," in *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, Oct. 2020, pp. 5252–5259.

[5] H. Yin, A. Varava, and D. Kragic, "Modeling, Learning, Perception, and Control Methods for Deformable Object Manipulation," in *Sci. Rob.*, vol. 6, May 2021, pp. 1–16.

[6] M. Yu, H. Zhong, and X. Li, "Shape Control of Deformable Linear Objects with Offline and Online Learning of Local Linear Deformation Models," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 1337–1343.

[7] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic Cable Routing with Spatial Representation," in *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, Apr. 2022, pp. 5687–5694.

[8] B. Lu, H. K. Chu, and L. Cheng, "Dynamic Trajectory Planning for Robotic Knot Tying," in *IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, 2016, pp. 180–185.

[9] V. Viswanath, J. Grannen, P. Sundaresan, B. Thananjeyan, A. Balakrishna, E. Novoseller, J. Ichnowski, M. Laskey, J. E. Gonzalez, and K. Goldberg, "Disentangling Dense Multi-Cable Knots," in *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, 2021, pp. 3731–3738.

[10] A. Keipour, M. Bandari, and S. Schaal, "Efficient Spatial Representation and Routing of Deformable One-Dimensional Objects for Manipulation," *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pp. 211–216, 2022.

[11] J. Guo, J. Zhang, D. Wu, Y. Gai, and K. Chen, "An Algorithm Based on Bidirectional Searching and Geometric Constrained Sampling for Automatic Manipulation Planning in Aircraft Cable Assembly," *J. Manuf. Syst.*, vol. 57, pp. 158–168, 2020.

[12] P. Chang and T. Padır, "Model-Based Manipulation of Linear Flexible Objects: Task Automation in Simulation and Real World," *Machines*, vol. 8, 2020.

[13] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking Deformable Objects with Point Clouds," *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 1130–1137, 2013.

[14] T. Tang, C. Wang, and M. Tomizuka, "A Framework for Manipulating Deformable Linear Objects by Coherent Point Drift," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3426–3433, 2018.

[15] T. Tang and M. Tomizuka, "Track Deformable Objects from Point Clouds with Structure Preserved Registration," *Int. J. Robot. Res.*, vol. 41, no. 6, pp. 599–614, 2022.

[16] C. Chi and D. Berenson, "Occlusion-Robust Deformable Object Tracking Without Physics Simulation," in *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, 2019, pp. 6443–6450.

[17] Y. Wang, D. McConachie, and D. Berenson, "Tracking Partially-Occluded Deformable Objects while Enforcing Geometric Constraints," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 14 199–14 205.

[18] Y. Yang, J. A. Stork, and T. Stoyanov, "Learning to Propagate Interaction Effects for Modeling Deformable Linear Objects Dynamics," *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pp. 4056–4062, 2021.

[19] W. Zhang, K. Schmeckpeper, P. Chaudhari, and K. Daniilidis, "Deformable Linear Object Prediction Using Locally Linear Latent Dynamics," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, June 2021, pp. 13 503–13 509.

[20] A. L. Yuille and N. M. Grzywacz, "A Mathematical Analysis of the Motion Coherence Theory," *Int. J. Comput. Vis.*, vol. 3, no. 2, pp. 155–175, 1989.

[21] A. Myronenko, X. Song, and M. Carreira-Perpiñá, "Non-Rigid Point Set Registration: Coherent Point Drift," *Adv. Neur. Inf. Proc. (NeurIPS)*, pp. 1–8, 2006.

[22] A. Myronenko and X. Song, "Point Set Registration: Coherent Point Drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, 2010.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data Via the *EM* Algorithm," *J. R. Stat. Soc., Ser. B, Methodol.*, vol. 39, pp. 1–22, 1977.

[24] S. Ge, G. Fan, and M. Ding, "Non-Rigid Point Set Registration with Global-Local Topology Preservation," *IEEE/CVF Int. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, pp. 245–251, 2014.

[25] Y. Yang, J. A. Stork, and T. Stoyanov, "Particle Filters in Latent Space for Robust Deformable Linear Object Tracking," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12 577–12 584, 2022.

[26] K. Lv, M. Yu, Y. Pu, X. Jiang, G. Huang, and X. Li, "Learning to Estimate 3-D States of Deformable Linear Objects from Single-Frame Occluded Point Clouds," in *IEEE Int. Conf. Robot. Autom. (ICRA) Workshop on Representing and Manipulating Deformable Objects*, May 2023.

[27] A. Keipour, M. Bandari, and S. Schaal, "Deformable One-Dimensional Object Detection for Routing and Manipulation," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4329–4336, 2022.

[28] R. Zanella, A. Caporali, K. Tadaka, D. De Gregorio, and G. Palli, "Auto-Generated Wires Dataset for Semantic Segmentation with Domain Independence," in *IEEE Int. Conf. Comput. Cont. Robot. (ICCCR)*. IEEE, Jan. 2021, pp. 292–298.

[29] A. Caporali, K. Galassi, G. Laudante, G. Palli, and S. Pirozzi, "Combining Vision and Tactile Data for Cable Grasping," in *IEEE/ASME Int. Conf. Adv. Intell. Mech. (AIM)*. IEEE, July 2021, pp. 436–441.

[30] A. Caporali, R. Zanella, D. De Gregorio, and G. Palli, "Ariadne+: Deep Learning-Based Augmented Framework for the Instance Segmentation of Wires," in *IEEE Trans. Ind. Inf.*, February 2022, pp. 1–11.

[31] H. Dinkel, J. Xiang, H. Zhao, B. Coltin, T. Smith, and T. Bretl, "Wire Point Cloud Instance Segmentation from RGBD Imagery with Mask R-CNN," in *IEEE Int. Conf. Robot. Autom. (ICRA) Workshop on Representing and Manipulating Deformable Objects*, May 2022.

[32] J. Xiang and H. Dinkel, "Simultaneous Shape Tracking of Multiple Deformable Linear Objects with Global-Local Topology Preservation," in *IEEE Int. Conf. Robot. Autom. (ICRA) Workshop on Representing and Manipulating Deformable Objects*, May 2023.

[33] M. Yu, K. Lv, C. Wang, M. Tomizuka, and X. Li, "A Coarse-to-Fine Framework for Dual-Arm Manipulation of Deformable Linear Objects with Whole-Body Obstacle Avoidance," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023.

[34] H. Dinkel, R. Navaratna, J. Xiang, B. Coltin, T. Smith, and T. Bretl, "Knot Tying with Virtual Fixtures," in *IEEE Int. Conf. Robot. Autom. (ICRA) Workshop on 3D Visual Representations for Robot Manipulation*, May 2024.

[35] S. A. I. Laboratory, "Robotic Operating System: Noetic Ninjemys," https://www.ros.org, 2018.

[36] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[37] C. R. Harris, J. Millman, S. van der Walt *et al.*, "Array Programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[38] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, 2007.

[39] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A Modern Library for 3D Data Processing," *arXiv:1801.09847*, 2018.

[40] P. Virtan, R. Gommers, T. E. Oliphant *et al.*, "Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nat. Methods*, vol. 17, pp. 261—-272, 2020.

[41] G. Guennebaud *et al.*, "Eigen v3," http://eigen.tuxfamily.org, 2010.

[42] R. B. Rusu and S. Cousins, "3D is Here: Point Cloud Library (PCL)," in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2011, pp. 1–4.