Simultaneous Shape Tracking of Multiple Deformable Linear Objects with Global-Local Topology Preservation

Jingyi Xiang[1], Holly Dinkel[2]

***Abstract*— This work presents an algorithm for tracking the shape of multiple entangling Deformable Linear Objects (DLOs) from a sequence of RGB-D images. This algorithm runs in real-time and improves on previous single-DLO tracking approaches by enabling tracking of multiple objects. This is achieved using Global-Local Topology Preservation (GLTP). This work uses the geodesic distance in GLTP to define the distance between separate objects and the distance between different parts of the same object. Tracking multiple entangling DLOs is demonstrated experimentally. The source code is publicly released.**

I. Introduction

Consider an automated robotic system which monitors in real-time the shape of a deformable linear object (DLO), for example a rope, a wire, or a string. This system could perceive the DLO in RGB-D imagery and estimate its configuration to perform a closed-loop manipulation task such as shape control or wire routing, or it could monitor the DLO for collision prevention [1]–[6]. These tasks are common in applications like robotic surgery, industrial automation, power line avoidance and human habitat maintenance. Previous work used physics simulation, including Finite Element Method (FEM) analysis, multi-physics, and dynamics, to model DLO motion [7]–[9], and motion planning frameworks predict minimal-energy wire configurations using the wire tips as boundary conditions [10]–[12]. This work builds on existing tracking methods through tracking multiple deforming and entangling DLOs simultaneously. This work make the following contributions:

1) We propose a method of tracking the shape of multiple DLOs in real-time. This is achieved by using the geodesic distance in the kernel describing how pairs of nodes influence each other's motion and setting the distance between independent objects to infinity.
2) We demonstrate tracking of multiple DLOs without instance segmentation in each frame. This is achieved by performing instance segmentation (expensive) on the first frame for initialization and performing semantic segmentation (cheap) on subsequent frames.
3) We release the source code and demonstration data at: https://github.com/RMDLO/multi-dlo.

[1]Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 61801. {jingyix4}@illinois.edu.

[2]Department of Aerospace Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 61801. {hdinkel2}@illinois.edu.

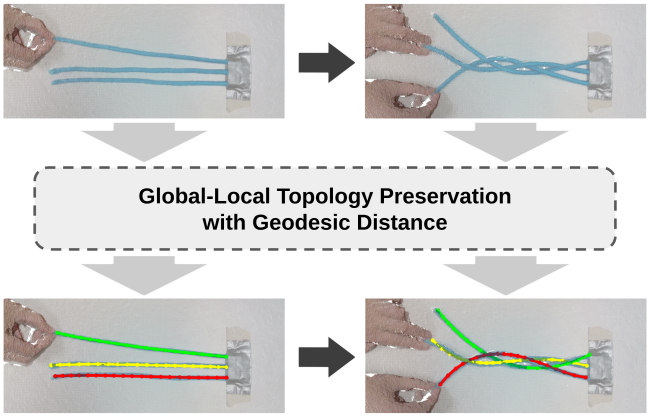

Fig. 1. Given a sequence of frames, Global-Local Topology Preservation tracks the shape of multiple deformable linear objects simultaneously.

II. Related Work

The Global-Local Topology Preservation (GLTP) algorithm performs modified non-rigid point set registration to map one set of points onto another using the Expectation-Maximization (EM) framework proposed by the Coherent Point Drift (CPD) algorithm [13]–[15]. The objective function for EM in GLTP uses CPD with locally linear embedding to preserve local topology. Non-rigid point set registration from CPD and GLTP set the foundation for several algorithms which perform DLO tracking under occlusion, including CPD+Physics, Structure Preserved Registration, CDCPD, CDCPD2, and TrackDLO [16]–[21]. These methods only demonstrate tracking one DLO at a time.

III. Methods

For N *points* in $\mathbb{R}^3$ received at time t from a depth sensor, $\mathbf{X}^t_{N\times 3} = (\mathbf{x}^t_1, \ldots, \mathbf{x}^t_N)^T$, the DLO shape can be represented by a collection of M ordered *nodes*, $\mathbf{Y}^t_{M\times 3} = (\mathbf{y}^t_1, \ldots, \mathbf{y}^t_M)^T$. The shape of each of K DLOs in a scene is represented by a different set of nodes, $\mathbf{Y}^t_{k\in[1,K]}$, each consisting of M_k nodes. Stacking all the $\mathbf{Y}^t_k$ matrices vertically produces $\mathbf{Y}^t$ with shape $M \times 3$, where $M = \sum_{k=1}^{K} M_k$. All K DLOs are treated as one combined object in the tracking process. From the tracking output $\mathbf{Y}^t$, the individual DLO nodes are obtained by accessing the corresponding rows in $\mathbf{Y}^t$.

Single DLO shape tracking algorithms generally perform object instance segmentation (conventionally, through color thresholding) in the RGB image first, and then segment the point cloud based on the RGB image segmentation [19], [20]. Our approach only requires instance segmentation for the first frame to initialize $\mathbf{Y}^t$. By treating all DLOs as one

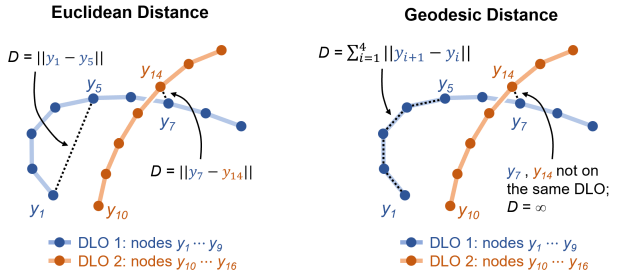

Fig. 2. Using Euclidean distance, the distance between nodes is the Euclidean norm of their difference. Using geodesic distance, the distance between two nodes on the same DLO is set to the sum of the segment lengths between them. If the two nodes are not on the same DLO, the distance between them is set to infinity.

deformable object, only semantic segmentation is required in subsequent frames. This approach bypasses the limitation of DLO instance segmentation in 2D RGB images, which can be slow and perform poorly [22]–[25].

A. Gaussian Mixture Model (GMM) Node Registration

Tracking begins with Gaussian Mixture Model (GMM) clustering as performed in [16]. The GMM clustering step computes $\mathbf{Y}^t$ as the centroids of Gaussian distributions from which $\mathbf{X}^t$ points are randomly sampled with isotropic variance σ^2. Assuming equal membership probability $p(m) = \frac{1}{M}$ and μ percent of the points are outliers, the GMM cost function takes the form

$$\begin{aligned} &E_{\text{GMM}}(\mathbf{y}_m^t, \sigma^2) \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M} p(m|\mathbf{x}_n^t)\frac{\|\mathbf{x}_n^t - \mathbf{y}_m^t\|^2}{2\sigma^2} + \frac{3N_p}{2}\log(\sigma^2), \end{aligned} \tag{1}$$

where

$$N_p = \sum_{n=1}^{N}\sum_{m=1}^{M} p(m|\mathbf{x}_n^t) \tag{2}$$

$$p(m|\mathbf{x}_n^t) = \frac{\exp\left(\frac{-D^2_{\mathbf{y}_m^t,\mathbf{x}_n^t}}{2\sigma^2}\right)}{\sum_{m=1}^{M}\exp\left(\frac{-D^2_{\mathbf{y}_m^t,\mathbf{x}_n^t}}{2\sigma^2}\right) + \frac{(2\pi\sigma^2)^{3/2}\mu M}{(1-\mu)N}} \tag{3}$$

and D is a distance metric (discussed in Section III-E).

B. The Motion Coherence Theory (MCT)

Given node positions $\mathbf{Y}^{t-1}$ and $\mathbf{Y}^t$ from consecutive time steps, the MCT defines a spatial velocity field $v(\mathbf{Y}^{t-1}) = \mathbf{Y}^t - \mathbf{Y}^{t-1}$ [26]. Nodes close to each other should move coherently through the smoothest possible spatial velocity field. For a spatial domain variable $\mathbf{z}$ and a frequency domain variable $\mathbf{s}$, the smoothness of the velocity field $v(\mathbf{z})$ can be measured by passing it through a high pass filter $1/\tilde{G}(\|\mathbf{s}\|)$ in the frequency domain as

$$E_{\text{MCT}}(v(\mathbf{z})) = \int_{\mathbb{R}^3} |\tilde{v}(\mathbf{s})|^2/\tilde{G}(\|\mathbf{s}\|)d\mathbf{s}, \tag{4}$$

where $\tilde{v}(\mathbf{s})$ and $\tilde{G}(\|\mathbf{s}\|)$ are the Fourier Transforms of $v(\mathbf{z})$ and $G(\|\mathbf{z}\|)$. By choosing $G(\|\mathbf{z}\|) = \exp(-\frac{\|z\|^2}{2\beta^2})$, this cost term is equivalent to the cost term of the MCT. The parameter β controls the width of high pass filter $1/\tilde{G}(\|\mathbf{s}\|)$. Therefore, a larger β leads to a smoother velocity field.

C. Locally Linear Embedding (LLE)

Locally Linear Embedding (LLE) represents a node $\mathbf{y}_m^{t-1}$ with its closest $2Q$ neighbors and a set of weights $\mathbf{L}$. At the next time step t, LLE preserves local topology by reconstructing $\mathbf{y}_m^t$ with its new neighbors and $\mathbf{L}$:

$$E_{LLE}(\mathbf{y}_m^t) = \sum_{m=1}^{M}\|\mathbf{y}_m^t - \sum_{i=m-Q}^{m+Q}\mathbf{L}(m,i)\mathbf{y}_i^t\|^2. \tag{5}$$

The weights $\mathbf{L}$ are computed with methods outlined in [27].

D. Expectation-Maximization Update for GLTP

The total cost of GLTP is updated iteratively using the EM algorithm. With $v(\mathbf{z})$ encoded in $\mathbf{y}_m^t$ as $\mathbf{y}_m^t = \mathbf{y}_m^{t-1} + v(\mathbf{y}_m^{t-1})$, the total cost is

$$\begin{aligned} &E(v(\mathbf{z}), \sigma^2) = E_{\text{GMM}} + E_{\text{MCT}} + E_{\text{LLE}} \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M}\frac{1}{2\sigma^2}p(m|\mathbf{x}_n^t)\|\mathbf{x}_n^t - \mathbf{y}_m^t\|^2 + \frac{3N_p}{2}\log(\sigma^2) \\ &\quad + \frac{\lambda}{2}\int_{\mathbb{R}^3}|\tilde{v}(\mathbf{s})|^2/\tilde{G}(\|\mathbf{s}\|)d\mathbf{s} \\ &\quad + \sum_{m=1}^{M}\|\mathbf{y}_m^t - \sum_{i=m-Q}^{m+Q}\mathbf{L}(m,i)\mathbf{y}_i^t\|^2 \end{aligned} \tag{6}$$

The solution to the above cost function takes the form [14]

$$v(\mathbf{z}) = \sum_{m=1}^{M}\mathbf{w}_m G(\|\mathbf{z} - \mathbf{y}_m\|). \tag{7}$$

For readability, we denote $\mathbf{X}^t$ as $\mathbf{X}$, $\mathbf{Y}^{t-1}$ as $\mathbf{Y}_0$, and use the following notations to solve for $\mathbf{w}_m$ and σ^2 analytically through EM:

- $\mathbf{W}_{M\times 3} = (\mathbf{w}_1^t, \ldots, \mathbf{w}_M^t)^T$, kernel weights,
- $\mathbf{P}_{M\times N}$, posterior matrix with $\mathbf{P}(m,n) = p(m|\mathbf{x}_n^t)$,
- $\mathbf{G}_{M\times M}$, kernel matrix with $\mathbf{G}(i,m) = \exp(-\frac{D^2_{\mathbf{y}_i^t,\mathbf{y}_m^t}}{2\beta^2})$
- $\mathbf{H}_{M\times M} = (\mathbf{I}-\mathbf{L})^T(\mathbf{I}-\mathbf{L})$, LLE weights,
- $\mathrm{d}(\mathbf{a})$, the diagonal matrix constructed from vector $\mathbf{a}$,
- $\mathrm{tr}(\mathbf{m})$, the trace of matrix $\mathbf{m}$, and
- $\mathbf{1}$, a column vector of ones.

The solutions $\mathbf{W}$ and σ^2 are computed from taking the partial derivatives $\frac{\partial E}{\partial \mathbf{W}}$ and $\frac{\partial E}{\partial \sigma^2}$ and setting them to zero as

$$\begin{aligned} \mathbf{W} = &(\mathrm{d}(\mathbf{P1})\mathbf{G} + \lambda\sigma^2\mathbf{I} + \alpha\sigma^2\mathbf{HG})^{-1} \\ &\cdot(\mathbf{PX} - (\mathrm{d}(\mathbf{P1}) + \alpha\sigma^2\mathbf{H})\mathbf{Y}_0) \end{aligned} \tag{8}$$

$$\begin{aligned} \sigma^2 = &\frac{1}{\mathbf{1}^T\mathbf{P1}D}(\mathrm{tr}(\mathbf{X}^T\mathrm{d}(\mathbf{P}^T\mathbf{1})\mathbf{X}) - 2\mathrm{tr}((\mathbf{PX})^T\mathbf{Y}_0) \\ &+ \mathrm{tr}(\mathbf{Y}_0^T\mathrm{d}(\mathbf{P1})\mathbf{Y}_0) + \mathrm{tr}(\mathbf{W}^T\mathbf{G}^T\mathrm{d}(\mathbf{P1})\mathbf{GW}) \cdot \\ &+ 2\mathrm{tr}(\mathbf{W}^T\mathbf{G}^T\mathrm{d}(\mathbf{P1})\mathbf{Y}_0) - \mathrm{tr}(\mathbf{W}^T\mathbf{G}^T\mathbf{PX})) \end{aligned} \tag{9}$$

The new node positions are $\mathbf{Y}^t = \mathbf{Y}^{t-1} + \mathbf{GW}$.

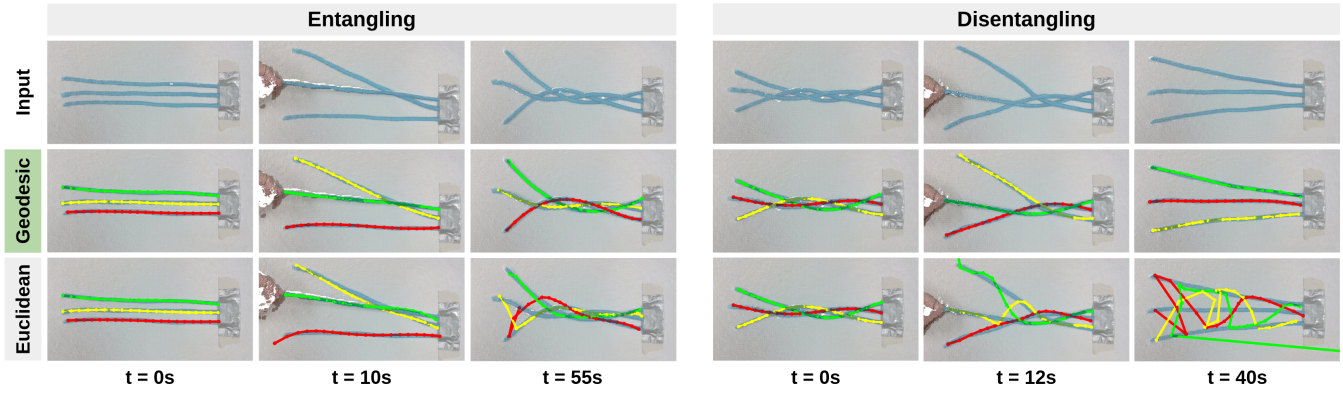

Fig. 3. Simultaneously tracking multiple DLOs with GLTP and geodesic proximity accurately estimates the shape of each DLO in demonstrations. By comparison, GLTP with Euclidean proximity fails to accurately track DLO shape in both the *Winding* and *Unwinding* scenarios.

E. Geodesic Proximity

The MCT requires the node velocity field to be smooth, where nodes spatially close to each other move in similar directions with similar speeds. In the node velocity given by

$$v(\mathbf{y}_m) = \sum_{i=1}^{M} \mathbf{G}(m,i)\mathbf{W}(i,\cdot) = \sum_{i=1}^{M} G(D_{\mathbf{y}_i,\mathbf{y}_m})\mathbf{W}(i,\cdot), \quad (10)$$

$D_{\mathbf{y}_i,\mathbf{y}_m}$ is the distance between nodes $\mathbf{y}_i$ and $\mathbf{y}_m$ given some distance metric D. Given G is Gaussian, $G(D_{\mathbf{y}_i,\mathbf{y}_m})$ is small if $\mathbf{y}_i$ is far from $\mathbf{y}_m$. This produces a small $G(D_{\mathbf{y}_i,\mathbf{y}_m})\mathbf{W}(i,\cdot)$, indicating the movement of $\mathbf{y}_i$ has insignificant influence on that of $\mathbf{y}_m$.

The Euclidean distance $d_{\mathbf{y}_i,\mathbf{y}_m} = \|\mathbf{y}_m - \mathbf{y}_i\|$ is a common choice for $D_{\mathbf{y}_i,\mathbf{y}_m}$, however it is not the best choice for representing the geometry of DLOs. If one DLO is resting on top of another DLO, the movement of the top one should have little influence on the movement of the bottom one. However, the small Euclidean distance between nodes near the intersection couples their motion together, causing tracking failure as shown in Fig. 3.

To resolve this, we use geodesic distance for D and define the node-to-node geodesic distance $\rho_{\mathbf{y}_i,\mathbf{y}_m}$ to be

$$\rho_{\mathbf{y}_i,\mathbf{y}_m} = \begin{cases} \sum_{j=m}^{i-1} \|\mathbf{y}_{j+1} - \mathbf{y}_j\| & \text{if } m \leq i \\ \sum_{j=i}^{m-1} \|\mathbf{y}_{j+1} - \mathbf{y}_j\| & \text{if } m > i \end{cases}. \quad (11)$$

If $\mathbf{y}_i$ and $\mathbf{y}_m$ are from two different DLOs, the distance between them is set to infinity.

Similarly, two nodes close in Euclidean distance to an intersection but from different DLOs should have different proximities to points near the intersection. Where $\mathbf{y}_{c_1}$ and $\mathbf{y}_{c_2}$ are the two nodes closest to point $\mathbf{x}_n$, we define the node-to-point geodesic distance $\rho_{\mathbf{y}_m,\mathbf{x}_n}$ as

$$\rho_{\mathbf{y}_m,\mathbf{x}_n} = \begin{cases} d_{\mathbf{y}_{c_1},\mathbf{x}_n} + \rho_{\mathbf{y}_{c_1},\mathbf{y}_m} & \text{if } \rho_{\mathbf{y}_m,\mathbf{y}_{c_1}} \leq \rho_{\mathbf{y}_m,\mathbf{y}_{c_2}} \\ d_{\mathbf{y}_{c_2},\mathbf{x}_n} + \rho_{\mathbf{y}_{c_1},\mathbf{y}_m} & \text{if } \rho_{\mathbf{y}_m,\mathbf{y}_{c_1}} > \rho_{\mathbf{y}_m,\mathbf{y}_{c_2}} \\ d_{\mathbf{y}_m,\mathbf{x}_n} & \text{if } m \in \{c_1, c_2\} \end{cases}. \quad (12)$$

IV. Results

We qualitatively demonstrate simultaneous multiple DLO shape tracking with intersection among independent objects. For these demonstrations, we ran tracking with $Q = 3, \beta = 0.8, \lambda = 1, \alpha = 3$, and an optimization function tolerance of $\epsilon = 10^{-5}$. We show performance in two scenarios:

1) *Entangling*–Three ropes lie parallel on a table. The ropes are crossed over each other and all three become twisted. This scenario demonstrates tracking when DLOs are wound, knotted, or tied.
2) *Disentangling*–Three ropes lie in a twisted configuration on a table. The ropes are uncrossed until all three ropes reach a separated, parallel configuration. This scenario demonstrates tracking when DLOs are unwound, unknotted, or untied.

The raw data comprise point cloud and RGB image data collected as the ropes are first wound and then unwound. The data are saved in one Robot Operating System (ROS) bag file. For initialization in both scenarios, instance segmentation was performed manually on the first frame due to the limitations of existing DLO instance segmentation algorithms [24], [25], [28]. The subsequent frames used color thresholding as the semantic segmentation input. The tracking results for these scenarios shown in Figure 3 highlight the accuracy of geodesic GLTP for multi-DLO shape tracking and the failure of Euclidean GLTP for this problem.

V. Conclusions and Limitations

We introduced a real-time, accurate algorithm for tracking multiple DLOs. One limitation of the method proposed is that a DLO could penetrate itself or other DLOs in the tracking result, which would not happen in real world situations. Potential ways to resolve this include incorporating physics simulators as described in Structure Preserved Registration or adapting the self-intersection constraint introduced in CDCPD2 [18], [20].

Acknowledgements

The authors thank the members of the Representing and Manipulating Deformable Linear Objects project (https://github.com/RMDLO) and the teams developing and maintaining the open-source software used in this project [29]–[36]. This work was supported by the Illinois Space Grant Consortium and the NASA Space Technology Graduate Research Opportunity 80NSSC21K1292.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] M. Ruan, D. M^{c}Conachie, and D. Berenson, "Accounting for Directional Rigidity and Constraints in Control for Manipulation of Deformable Objects without Physical Simulation," *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, pp. 512–519, 2018.

[9] 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.

[10] M. Moll and L. Kavraki, "Path Planning for Deformable Linear Objects," *IEEE Trans. Robot.*, vol. 22, pp. 625–636, 2006.

[11] T. Bretl and Z. McCarthy, "Mechanics and Quasi-Static Manipulation of Planar Elastic Kinematic Chains," *IEEE Trans. Robot.*, vol. 29, pp. 1–14, 2012.

[12] ——, "Quasi-Static Manipulation of a Kirchoff Elastic Rod Based on a Geometric Analysis of Equilibrium Configurations," *Int. J. Robot. Res.*, vol. 33, pp. 48–68, 2013.

[13] 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.

[14] 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.

[15] 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.

[16] T. Tang, Y. Fan, H.-C. Lin, and M. Tomizuka, "State Estimation for Deformable Objects by Point Registration and Dynamic Simulation," in *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, 2017, pp. 2427–2433.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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 *Under Review*, 2023, pp. 1–8.

[22] 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.

[23] 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.

[24] H. Dinkel, H. Zhao, J. Xiang, B. Coltin, T. Smith, and T. Bretl, "Benchmarking Wire Instance Segmentation with Mask R-CNN," in *Under Submission*, 2023.

[25] 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/RSJ Int. Conf. Robot. Autom. (ICRA) Workshop on Deformable Objects*, May 2022.

[26] 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.

[27] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, "Locally linear embedding and its variants: Tutorial and survey," *arXiv preprint arXiv:2011.10925*, 2020.

[28] 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.

[29] Stanford Artificial Intelligence Laboratory, "Robotic Operating System: Noetic Ninjemys," 2018. [Online]. Available: https://www.ros.org

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

[31] C. R. Harris, J. Millman, S. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array Programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

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

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

[34] P. Virtan, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, "Scipy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nat. Methods*, vol. 17, pp. 261—-272, 2020.

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

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