KnotDLO: Toward Interpretable Knot Tying

Holly Dinkel[1,3], Raghavendra Navaratna[1], Jingyi Xiang[2], Brian Coltin[3], Trey Smith[3], Timothy Bretl[1]

***Abstract*— This work presents KnotDLO, a method for one-handed knot manipulation to transform a Deformable Linear Object (DLO) into knot states. Grasp and target waypoints for future DLO states are planned from the current DLO shape. Grasp poses are computed from indexing the tracked piecewise linear curve representing the DLO state based on the current curve shape and are piecewise continuous. Intermediate waypoints are computed from the geometry of the current DLO state and the desired next state. The perception and manipulation system is robust to occlusion, repeatable for varying rope initial configurations, interpretable for generating motion policies, and requires no human demonstrations or training. The system decouples visual reasoning from control. In 16 trials of knot tying, KnotDLO achieves a 50% success rate in tying an overhand knot from previously unseen configurations.**

I. Introduction

Manipulating Deformable Linear Objects (DLOs) into knots is useful for many robotic tasks, including climbing [1], [2], surgery [3]–[6], household robotics [7], and industrial robotics [8]–[10]. Manipulating DLOs can be categorized based on the type of DLO shape information used as input. Tasks such as insertion [11], disentangling [7], [12], and clearing DLOs from the path to other objects [13] may only require information such as the location of the tips, the locations of crossings, or the locations of inner and external knot segments rather than a full shape estimate. Tasks such as winding, braiding, and knotting may benefit from a complete shape estimate [14]. One challenge is representing the DLO topology, especially when the DLO is occluded by itself or the environment as it is manipulated. Recent work in real-time visual DLO tracking enables occlusion-robust topology representation for autonomous knot tying. KnotDLO achieves repeatable knot tying using topological waypoints validated in overhand knot tying experiments with a success rate of 50%, compared to the success rate of the learning-based state-of-the-art method of 66% [15]. The topological waypoints enable transformation between knot states using movement primitives independent of task parameterizations such as initial DLO configuration or DLO length. Geometric grasp and target poses are computed from the geometry of the DLO and are a subset of a continuous space of grasp and target poses defined by the DLO topology.

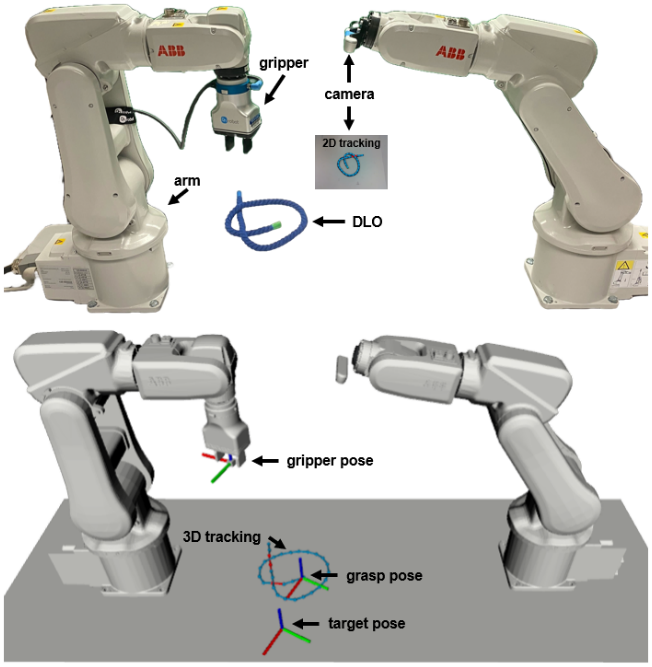

Fig. 1. One robot arm with a parallel gripper grasps the DLO and moves it between topological waypoints. The DLO topology is tracked from imagery collected by a camera mounted on the last link of a second robot arm.

II. Related Work

The availability of an accurate DLO topology for robotic knot tying has conventionally been limited by perception systems which struggle to follow the shape of the DLO as it is occluded within the environment or as it is occluded by itself during the formation of intersections. One method overcomes this by learning a fast and differentiable neural network model of the DLO dynamics to augment segmentation-based shape estimation with physics [16], [17]. Another method learns DLO configuration-independent dense depth object descriptors from synthetic ropes rendered in Blender and uses these descriptors to learn manipulation action sequences with visual imitation of demonstrations [15]. This method encodes geometric structure on the descriptors and learns them in simulation without any robot motion learning. This accelerates training time substantially from work which requires collecting robot motion demonstration data for imitation [18], [19] and enables transferring the learned visual representation across domains. Other work performs DLO shape manipulation with visual feedback for shape control without self-occlusion [17], [20], [21]. Recent advances in DLO perception enable DLO shape tracking under environment occlusion [22]–[25]. These methods also

[1] Department of Aerospace Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 61801. {hdinkel2, rsn3, tbretl}@illinois.edu.

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

[3] Intelligent Robotics Group, NASA Ames Research Center, Moffett Field, CA, 94035 USA {brian.coltin, trey.smith}@nasa.gov.

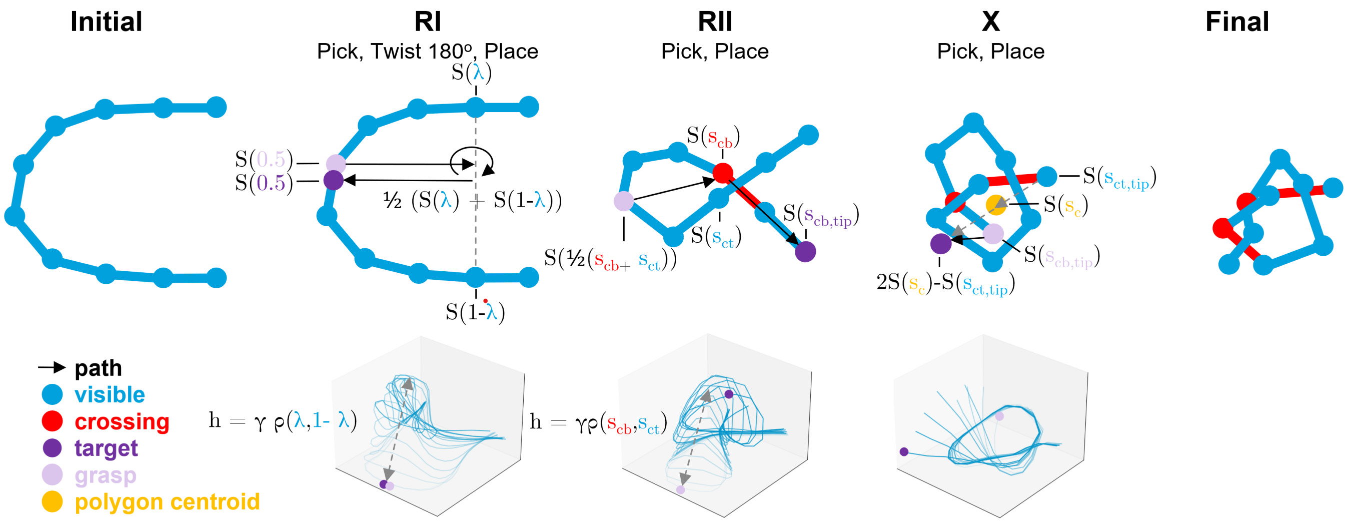

Fig. 2. Given an initial symmetric, curve-shaped DLO configuration, the planning system computes grasp and target indices from the topology to compose a sequence knot tying movement primitives. (Top) Sequentially performing Reidemeister Move I (RI), Reidemeister Move II (RII), and the Cross Move (X) results in an overhand knot. (Bottom) The tracking system tracks the shape of the DLO as it moves.

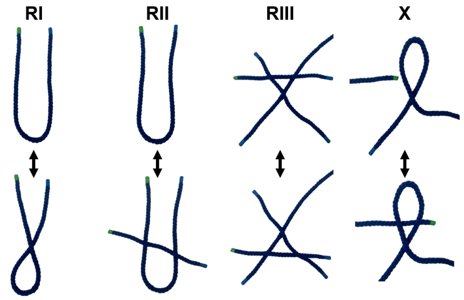

Fig. 3. Four movement primitives can transform a DLO between any knot states. (Left) Reidemeister Move I twists the DLO to add or remove one crossing. (Center left) Reidemeister Move II slides one strand over one loop to add or remove two crossings. (Center right) Reidemeister Move III slides one strand over a crossing to transform the knot state around the crossing. (Right) The Cross Move pokes the tip of a DLO through a loop to add or remove two crossings.

accurately track shape under self-occlusion, offering new opportunities for full DLO state representation for knot tying.

The Reidemeister moves are a set of movement primitives which add or remove links in a knot projection, creating a new projection. Reidemeister Move I (RI) adds or removes one crossing through twisting or untwisting. Reidemeister Move II (RII) adds or removes two crossings by sliding one straight segment over one curved segment. Reidemeister Move III (RIII) transforms the topology of a crossing by sliding one segment over or under a crossing [26]. The Cross Move (X) was introduced for tying objects which are not closed loops in the unknotted state. It passes the tip of the rope through a loop in the knot to create a crossing. Composing the RI, RII, and X moves in sequence results in an overhand knot.

III. One-Handed Overhand Knot Manipulation

This work presents an overhand knot tying system using real-time visual DLO shape tracking to compute manipulation waypoints [25]. The topology of a DLO is represented as a piecewise linear curve, S, constructed from a set of M control points, $C^{M \times 3} \in \mathbb{R}^3$. The curve S is parameterized at time t by curvilinear length, s^t, and total length, L, with

$$S : s \in [0,1] \rightarrow S(s^t; L) \in \mathbb{R}^3. \quad (1)$$

The head and tail positions of the DLO are $S(0)$ and $S(1)$. Two curvilinear lengths, s_i^t and s_j^t, are determined to be on the same segment of $S(s^t; L)$ if $|\text{idx}_i - \text{idx}_j| \leq 1$, where

$$\text{idx}_i = \lfloor s_i \times (M-1) \rceil. \quad (2)$$

The geodesic distance defines the length of the shortest path along a manifold. The geodesic distance, $\rho(s_i^t, s_j^t)$, is

$$\rho(s_i^t, s_j^t) = \begin{cases} \left|s_j^t - s_i^t\right| \cdot \left\|S(s_j^t) - S(s_i^t)\right\| \text{ if } |\text{idx}_j - \text{idx}_i| \leq 1 \\ \left|1 - s_j^t\right| \cdot \left\|S(s_i^t) - C^t_{\text{end,i}}\right\| + \left|s_j^t\right| \cdot \left\|S(s_j^t) - C^t_{\text{end,j}}\right\| \\ + \sum_{m=\text{end,i}+1}^{\text{end,j}} \left\|C_m - C_{m-1}\right\| \text{ otherwise} \end{cases}, \quad (3)$$

where $\|\cdot\|$ is the l_2 norm and

$$C^t_{\text{end,i}} = \underset{m \in M}{\text{argmin}} \left\|S(s_i^t) - C_m\right\|. \quad (4)$$

The DLO topology is used to plan grasp and target poses for a robot with a gripper to tie an overhand knot. As shown in Fig. 4, the orientation of these poses are computed using the semi-planar manipulation constraint (i.e., $z = [0,0,1]$), where the $y-$axis of the transform is the unit-length tangent vector to the curve at the desired point, $dS(s^t;L)/ds$, and the $x-$axis is the axis normal to the $y-z$ plane. Waypoints transforming each grasp pose to the target pose are designed to reduce effects from sliding on the manipulation plane. Seven curvilinear lengths—$s = 0.5$, $s = \lambda$, $s = 1 - \lambda$, s_{cb} (the crossing bottom), s_{ct} (the crossing top), $s_{\text{cb, tip}}$ (the undertip), and $s_{\text{ct, tip}}$ (the overtip)—are selected to tie an overhand knot for a DLO in an initial symmetric, curved configuration as shown in Fig. 2. The DLO is first grasped at $S(0.5)$, lifted to a height $h = \gamma\rho(\lambda, 1-\lambda)$, and translated to the $x-y$ positions given by the midpoint between the two parallel lengths, $\frac{1}{2}\left(S(\lambda) + S(1-\lambda)\right)$. At this waypoint, the grasped point is rotated by $\frac{\pi}{2}$ radians about the z-axis,

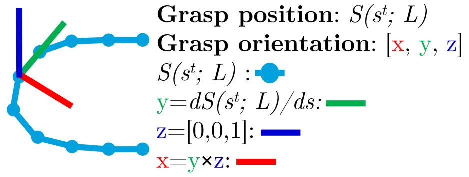

Fig. 4. A grasp pose can be computed given the curve representation of the tracking result, $S(s^t; L)$, and the curvilinear length at a given time step, s^t. The grasp position is $S(s^t)$ and the orientation is computed from the unit-length tangent vector to the curve, $dS(s^t; L)/ds$ assuming semi-planar knot tying with $z = [0, 0, 1]$.

TABLE I

KNOT TYING SUCCESS RATE BY MOVE

RI	RII	X	**Total**
0.937	0.867	0.615	**0.500**

then returned to the start position of the move to complete RI. The top crossing, s_{ct}, is calculated as the control point closest in Euclidean distance to the self-occluded control point estimated from the tracking algorithm using their 2D projections in pixel space [25]. The undertip is computed as the tip closest to the crossing bottom using Eq. (3). To perform RII, the DLO is grasped at $S(0.5)$, lifted to a height $h = \gamma\rho(s_{\text{cb}}, s_{\text{ct}})$, translated to $S(s_{\text{cb}})$, translated to $S(s_{\text{cb,tip}})$, and placed. Tracking often fails to accurately initialize the crossing topology after this step due to its sensitivity to depth resolution, however the X move requires grasping near $S(s_{\text{cb, tip}})$. Instead of computing $S(s_{\text{cb, tip}})$ from topology, $S(s_{\text{cb, tip}})$ is computed as the tip nearest to the centroid of a convex polygon formed from the tracked points along the DLO [27]. To improve robustness, the closest node to the polygon centroid within a radius of r nodes from the undertip is selected as the grasp point. The target position for the X move is computed as $S(s_c) + (S(s_c) - S(s_{ct,tip}))$ and the target orientation is the same as the grasp orientation.

IV. EXPERIMENTS

The DLO is assumed to start in a symmetric curved shape and able to be distinguished from the background using image-based thresholding in Hue, Saturation, and Value (HSV) color space. One tip of the rope is covered with green tape to break object symmetry when initializing the object shape. In 16 experiments analyzing the repeatability of one-handed knot manipulation, the system achieves a 50% success rate in tying an an overhand knot from previously unseen initial DLO configurations. All experiments used an Intel RealSense d435 camera for vision with default tracking parameters in TrackDLO with the number of nodes used as control points, M, as $M = 30$. The DLO is a blue rope with $L = 0.88\text{m}$, $\lambda = 0.1$, $\gamma = 0.4$ and $r = 5$. The manipulation system includes one ABB IRB120 manipulator with an OnRobot 2FG7 gripper. Manipulation assumes the perceived topology of the DLO is accurate before performing RI and RII, but not before performing RIII. Figure 5 shows the steps of knot tying in the manipulation scene, which shows the transformation between knot states, and the planning scene, which shows the poses of waypoints. Table I reports per-move knot tying success rates. Table II reports knot tying success rates from the literature.

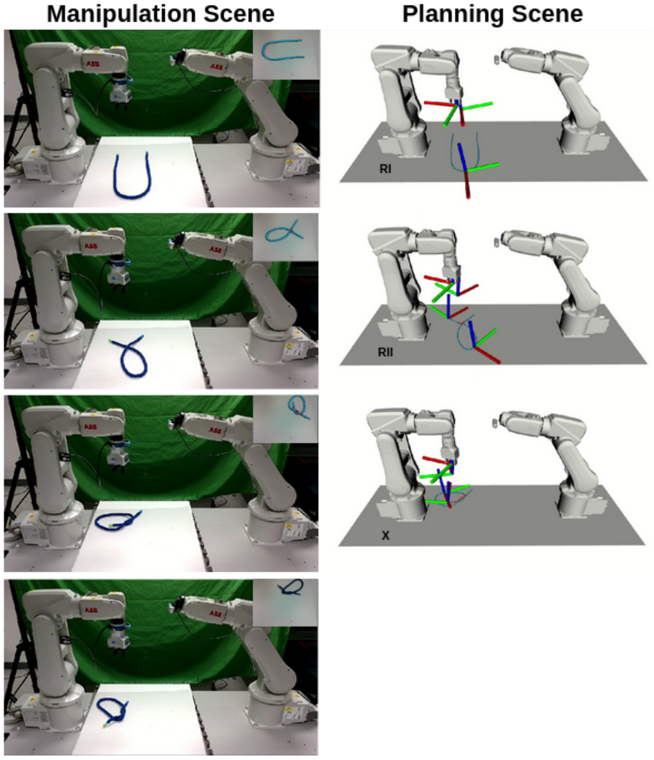

Fig. 5. A small number of waypoints designed based on the rope topology are required to tie an untightened overhand knot.

TABLE II

OVERALL OVERHAND KNOT TYING SUCCESS RATES

KnotDLO (Ours)	DDOD [15]	GSP [18]	Imitation [19]
0.50	**0.66**	0.6	0.38

V. CONCLUSIONS

This work introduced KnotDLO, a system for semi-planar one-handed overhand knot tying. KnotDLO is the first knot tying method using topological DLO state tracking from a perception system as input. The knot tying system was evaluated in repeated knot tying trials from different initial symmetric curved configurations. The key advantages of this system are its interpretability and robustness to occlusion, including its ability to resolve points of crossing in the DLO. This method shows promise for further development and deployment as an explainable system for knot tying. In future work, KnotDLO could also be used to automatically collect high-quality demonstration data to learn multi-step manipulation policies comprising sequences of pick-and-place actions [28].

ACKNOWLEDGEMENTS

The authors thank the members of the Representing and Manipulating Deformable Linear Objects project (https://github.com/RMDLO) and the teams developing the open-source software used in this project [27], [29]–[37]. This work was supported by the NASA Space Technology Graduate Research Opportunity 80NSSC21K1292, the P.E.O. Scholar Award, and the Zonta International Amelia Earhart Fellowship.

References

[1] J. J. Page, L. K. Treers, S. Jorgensen, R. Fearing, and H. S. Stuart, “The Robustness of Tether Friction in Non-Idealized Terrains,” *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 424–431, 2023.

[2] A. Specian and M. Yim, “Friction Binding Study and Remedy Design for Tethered Search and Rescue Robots,” in *IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR)*, 2015, pp. 1–6.

[3] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, “A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks,” in *IEEE/RSJ Int. Conf. Intell. Robot. Sys. (IROS)*, 2006, pp. 543–548.

[4] S. Wang, H. Wang, and L. Yue, “A Novel Knot-Tying Approach for Minimally Invasive Surgical Robot Systems,” *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 4, no. 3, pp. 268–276, 2008.

[5] H. Wang, S. Wang, J. Ding, and H. Luo, “Suturing and Tying Knots Assisted by a Surgical Robot System in Laryngeal MIS,” *Robotica*, vol. 28, no. 2, pp. 241–252, 2010.

[6] D.-L. Chow and W. Newman, “Improved Knot-Tying Methods for Autonomous Robot Surgery,” in *IEEE Int. Conf. Autom. Sci. Eng. (CASE)*. IEEE, 2013, pp. 461–465.

[7] 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.

[8] P. Mitrano, D. McConachie, and D. Berenson, “Learning Where to Trust Unreliable Models in an Unstructured World for Deformable Object Manipulation,” *Sci. Rob.*, vol. 6, no. 54, 2021.

[9] X. Li, X. Su, Y. Gao, and Y.-H. Liu, “Vision-Based Robotic Grasping and Manipulation of USB Wires,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2018, pp. 3482–3487.

[10] S. Jin, D. Romeres, A. Ragunathan, D. K. Jha, and M. Tomizuka, “Trajectory Optimization for Manipulation of Deformable Objects: Assembly of Belt Drive Units,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2021, pp. 10 002–10 008.

[11] W. Wang, D. Berenson, and D. Balkcom, “An Online Method for Tight-Tolerance Insertion Tasks for String and Rope,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2015, pp. 2488–2495.

[12] H. Wakamatsu, E. Arai, and S. Hirai, “Knotting/Unknotting Manipulation of Deformable Linear Objects,” *Int. J. Robot. Res.*, vol. 25, no. 4, pp. 371–395, 2006.

[13] D. Rojas, “Autonomous Robotic Manipulation of Deformable Linear Objects During Deep Space Maintenance and Repair Procedures,” Ph.D. dissertation, UC Davis, 2022.

[14] 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.

[15] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, “Learning Rope Manipulation Policies Using Dense Object Descriptors Trained on Synthetic Depth Data,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2020, pp. 9411–9418.

[16] 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.

[17] 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.

[18] D. Pathak, P. Mahmoudieh, G. Luo, P. Agrawal, D. Chen, Y. Shentu, E. Shelhamer, J. Malik, A. A. Efros, and T. Darrell, “Zero-Shot Visual Imitation,” in *Int. Conf. Learn. Represent.*, 2018.

[19] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, “Combining Self-Supervised Learning and Imitation for Vision-Based Rope Manipulation,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*. IEEE, 2017, pp. 2146–2153.

[20] 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.

[21] M. Yu, K. Lv, H. Zhong, S. Song, and X. Li, “Global Model Learning for Large Deformation Control of Elastic Deformable Linear Objects: An Efficient and Adaptive Approach,” *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 417–436, 2023.

[22] 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.

[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] Y. Wang, D. M^{c}Conachie, 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.

[25] 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. Autom. Lett.*, 2023, pp. 1–8.

[26] K. Reidemeister, *Knot Theory*. Springer-Verlag, 1932.

[27] S. Gillies, C. van der Wel, J. Van den Bossche, M. W. Taves, J. Arnott, B. C. Ward *et al.*, “Shapely,” Oct. 2022. [Online]. Available: https://github.com/shapely/shapely

[28] D. Seita, P. Florence, J. Tompson, E. Coumans, V. Sindhwani, K. Goldberg, and A. Zeng, “Learning to Rearrange Deformable Cables, Fabrics, and Bags with Goal-Conditioned Transporter Networks,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 4568–4575.

[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] K. Koide and E. Menegatti, “General Hand–Eye Calibration Based on Reprojection Error Minimization,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1021–1028, 2019.

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

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