Detection and Physical Interaction with Deformable Linear Objects

Azarakhsh Keipour[1], Mohammadreza Mousaei[2], Maryam Bandari[3], Stefan Schaal[4] and Sebastian Scherer[5]

***Abstract*— Deformable linear objects (e.g., cables, ropes, and threads) commonly appear in our everyday lives. However, perception of these objects and the study of physical interaction with them is still a growing area. There have already been successful methods to model and track deformable linear objects. However, the number of methods that can automatically extract the initial conditions in non-trivial situations for these methods has been limited, and they have been introduced to the community only recently. On the other hand, while physical interaction with these objects has been done with ground manipulators, there have not been any studies on physical interaction and manipulation of the deformable linear object with aerial robots.**

This workshop describes our recent work on detecting deformable linear objects, which uses the segmentation output of the existing methods to provide the initialization required by the tracking methods automatically. It works with crossings and can fill the gaps and occlusions in the segmentation and output the model desirable for physical interaction and simulation. Then we present our work on using the method for tasks such as routing and manipulation with the ground and aerial robots. We discuss our feasibility analysis on extending the physical interaction with these objects to aerial manipulation applications.

I. Introduction

Deformable One-dimensional Objects (DOOs) or Deformable Linear Objects (DLOs) are a class of objects that includes ropes, cables, threads, sutures, and wires. A crucial part of achieving full autonomy for physical interaction with DLOs is perception. Many applications require complete knowledge of the object's initial state as a model that allows simulation and the computation of its dynamics.

Many researchers in the medical and industrial community have proposed methods to extract DLOs from images [1], [2], [3], [4], [5]. These methods mainly provide the region in the image containing the DLO and are not directly suitable for autonomous physical interaction. On the other hand, various algorithms have been proposed to track a DLO across the video frames [6], [7], [8], [9], [10], [11], [12], [13], [14]. These methods have been used for physical interaction and manipulation with ground robots, and while some of these methods can detect the initial DLO state in trivial conditions (e.g., a straight rope in camera view), others require even a simple DLO configuration to be provided to them a priori. On the other hand, the aerial robotics community has extensively worked on the segmentation of cables and wires for visual inspection and obstacle avoidance purposes. These methods can effectively segment out the power lines, bridge cables, and other near-straight deformable objects [15], [16], [17], [18], [19], [20], [21], [22]. However, none of these approaches can handle true deformations, making these approaches unsuitable for physical interaction tasks.

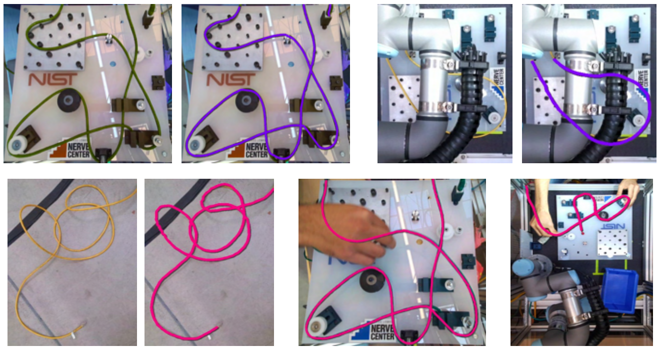


Fig. 1: The result of DLO detection (overlaid with purple and pink) on inputs with occlusions and crossings [24].

While providing the initial DLO state in the lab settings is possible, it is not generally provided in real-world applications. On the other hand, the existing learning-based detection methods are labor-intensive to train for each new DLO and new setting and are hard to generalize to less-certain conditions of the real-world [23].

This workshop briefly reviews our approach from [24] to detect the initial conditions of a deformable linear object in more complex scenarios, with gaps, occlusions, and DLO crossings. We further analyze our tested applications of this method in ground manipulation and routing and study the feasibility of tasks involving the physical interaction of DLOs for aerial robots.

II. The Method

Our detection method outputs the DLO state as a chain of fixed-sized segments connected by passive spherical joints, commonly used for manipulation and dynamic simulation. The method has six steps: segmentation, skeletonization, contour extraction, DLO fitting, pruning, and merging. The first three steps can use a combination of existing approaches, while the rest of the steps are specific to this method. The algorithm is briefly described in this section but is explained in more detail in [24].

A. Segmentation, Skeletonization and Contour Extraction

Segmentation filters the image data to extract the DLO portions and exclude all other data. Many model-based and

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA keipour@gmail.com

[2,5] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA [mmousaei, basti]@cmu.edu

[3,4] Intrinsic, Mountain View, CA [maryamb, sschaal]@intrinsic.ai

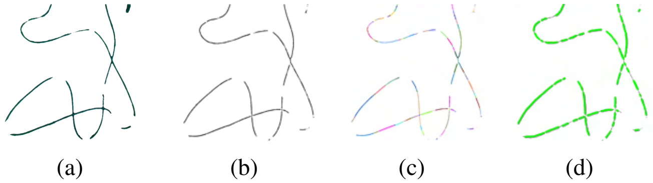


Fig. 2: The first four steps of DLO detection on an image frame from Figure 1 [24]. (a) Segmentation. (b) Skeletonization. (c) Contour extraction. (d) DLO Fitting and Pruning.

learning-based segmentation methods have been proposed in both medical and industrial robotics communities [1], [2], [3], [4], [5], [25], [26], [27]. This step should eliminate all the unrelated data, even if it removes some DLO data.

Skeletonization transforms each segmented connected component into a set of connected pixels with single-pixel width. It is commonly used in many applications [28], [29]. Our algorithm requires the skeleton of each connected component to remain connected and only one branch to be returned per actual branch. We used a well-known morphological thinning method for skeletonization [30].

Extracting contours is also a standard step in many applications [31], [32]. The contour extraction methods can result in several contours per branch and some with multiple branches. Our DLO detection method can handle these issues, and many of the existing contour extraction methods can be used with it. We used Suzuki and Abe's method [33] to extract contours.

Figures 2(a-c) present the results of the first three steps on an example camera frame.

B. DLO Fitting and Pruning

Each contour can be a single branch, or it may contain multiple branches. The pixel sequence for a contour returned by a typical contour extraction method starts from one of the branch tips, goes around the skeleton component, and ends with a sharp turn back at the start point.

Let us define the length of each DLO segment (i.e., the fixed cylinder) as l_s. The DLO fitting step initializes an empty DLO chain, starts from the first pixel in the contour, and traverses over the pixels until the distance from that pixel is l_s. It creates a new cylindrical segment, adds it to the chain, and continues traversing from the end of the new cylinder again. Every time it goes over a branch tip, it records the current DLO chain and initializes a new one. This continues until traversing reaches the last point in the contour sequence.

This method returns multiple overlapping DLO chains for each part of the object, and overlapping segments are pruned to simplify the further steps by assuming that no segments overlap. Figure 2(d)(d) shows the result of fitting and pruning.

C. Merging

With the collection of DLO chains, they are iteratively merged to fill the gaps and form a single object. Each iteration connects two chains, and the process is repeated until there is only one chain left. With two chains each having two ends, there are four cost values for connecting the two ends for any two chains. The lowest among the four values is the cost of connecting the two chains, and the chain pair with the lowest cost is connected at each iteration.

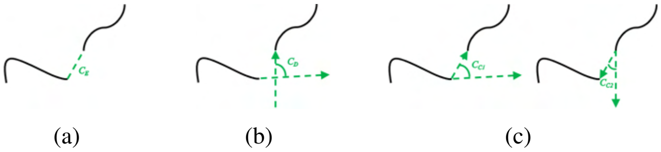


Fig. 3: Partial merging costs for two chain ends [24]. (a) Euclidean cost. (b) Direction cost. (c) Curvature costs.

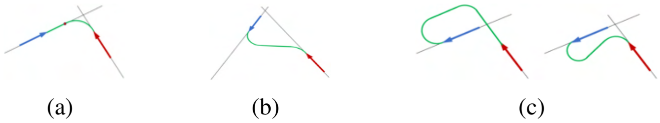


Fig. 4: Illustration of suggested solutions for merging scenarios [24]. The arrows represent the DLO chain ends, with the arrow sides representing the end points.

TABLE I: Detection results on 7 video sequences [24].

	Total	Correct	Incorrect	Accuracy
Frames	4,230	3,542	688	83.7%
Occlusions	26,456	23,991	2,465	90.7%
Merges	583,743	581,130	2,613	99.6%

Three separate partial costs are used for computing the total connection cost: the Euclidean distance of the two chain ends (Figure 3(a)), the difference in their directions (Figure 3(b)), and how much curvature is needed to connect them (Figure 3(c)).

The gap between the two chains should be filled to follow the expected curve of the deformable object. While any deformable object can take almost any curve, it is possible to have an educated *guess*. We calculate the "natural" curvature, which connects the desired ends of the two chains. The "natural" curvature initially follows the direction of the last segments of the chains and then follows a curve with the largest constant turn rate possible. Our suggested solutions for the different possible cases are shown in Figure 4.

III. Experiments and Results

Figure 1 shows the detection results on some input frames. The average detection time per frame across all the sequences is 0.537 seconds on a system with Intel® Core™ i9-10885H CPU and 64 GB DDR4 RAM and a sub-optimal implementation in Python 3. We tested the method on 7 video sequences with 4,230 frames of size 1280×720. Table I shows the quantitative results for the algorithm's accuracy on the whole cable in an image, for the occlusions filled, and for the merges performed. For a more detailed explanation of the results, please see [24].

IV. Feasibility Analysis for Manipulation

To test the feasibility of the detection method for fully-automated physical interaction, we have used it in a cable

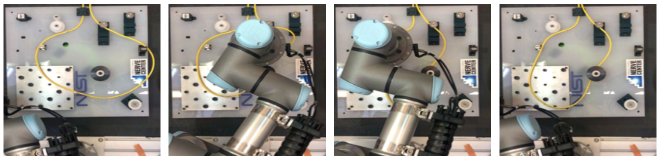
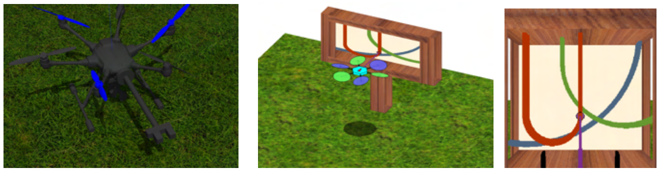


Fig. 5: Cable routing and manipulation using a UR3 robotic arm [34].



(a) (b)

Fig. 6: Feasibility tests of DLO manipulation using aerial robots in simulation. (a) Gazebo model of our robot with a gripper for cable manipulation. (b) MATLAB cable grasping tests.

TABLE II: Multirotor end-effector position error (in $[mm]$) for grasping a cable segment. Trials in Gazebo simulator.

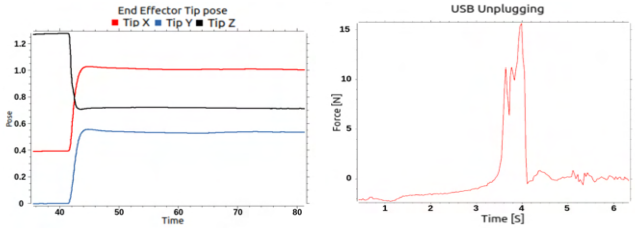
# of Tests	Max. Error	Mean Error	Std. Dev.
20	12.92	7.84	2.91

routing and manipulation application with a ground UR3 manipulator arm [34]. Figure 5 shows screenshots from one of the tests.

In general, ground manipulators have higher precision compared to aerial robots. For aerial DLO manipulation (for example, for maintenance and cable manipulation at the top of utility poles), a significant issue is the precision of the end-effector in grasping the DLOs detected in the camera. We tested the feasibility of manipulating the detected cable in Gazebo and MATLAB for a fully-actuated hexarotor with tilted arms controlled with the system introduced in [35]. We measured the position error for grasping a specific point on the cable [36]. Figure 6 shows our setup for testing the feasibility of the task. The error from the MATLAB simulator is near zero. The Gazebo simulator tends to give more realistic results, so we only report the Gazebo experiments.

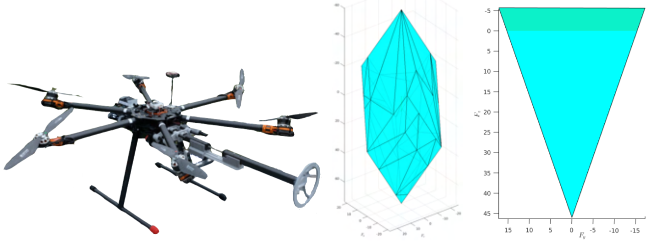
For each experiment, the robot first flies to around $0.5m$ distance from the cable, then moves forward to grasp the cable segment. Figure 7(a) illustrates how the end-effector's position can reach the target cable point. Table II shows the viability of the physical interaction with the perceived DLOs in the simulation if at least 13 $[mm]$ position error in grasping can be tolerated in the application. The next future step would be to perform the analysis on the real robot.

On the other hand, aerial robots have limited wrenches compared to ground robots. It is imperative to analyze the feasibility of the physical interaction tasks from the manipulability perspective as well. We computed the force polytopes (Figure 8(b)) for our aerial manipulator shown in Figure 8(a) (see [35]) and measured the forces required for simple cable-related tasks (such as un/plugging USB cables) [36].



(a) (b)

Fig. 7: Feasibility tests of aerial DLO manipulation in Gazebo. (a) End-effector's position to grasp the desired cable segment at [1.0, 0.5, 0.7]. (b) Forces required during unplugging a USB Type A cable.



(a) (b) (c)

Fig. 8: Feasibility tests of DLO manipulation forces. (a) Our aerial platform. (b) Force polytopes for our platform. (c) Remaining y and z forces when unplugging a USB cable.

Figure 7(b) shows the example forces measured for unplugging a USB Type A cable which in this scenario is $15.84\ [N]$ at the peak. Figure 8(c) shows the available forces for our robot when it is pulling the plug with $15.84\ [N]$ directly in its backward direction. The green region shows the remaining forces that allow the robot to keep its altitude. This analysis shows that our aerial robot would be able to unplug the cable in this case, but it is very close to its limits and may not be able to perform a more demanding task.

V. Future Work

The proposed DLO detection method is a good stepping stone to having a more comprehensive and powerful detection method, and several improvements can be made to the current version. Here are the potential future work directions for both the detection method and its applications:

- Detecting which DLO segment is on top of the other at the crossings.
- Combining detection with a powerful DLO segmentation and a tracking method to get the full perception system.
- Implementing the method more optimally for real-time applications.
- Extending and testing the detection method for 3-D.
- Performing the feasibility analysis on the real aerial robot.
- Test the detection method in a real-world aerial manipulation task.

REFERENCES

[1] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset, “Model-based detection of tubular structures in 3d images,” *Computer Vision and Image Understanding*, vol. 80, no. 2, pp. 130–171, 2000.

[2] T. Hachaj and M. R. Ogiela, “Segmentation and visualization of tubular structures in computed tomography angiography,” in *Intelligent Information and Database Systems*, 2012, pp. 495–503.

[3] B. Wang, H. Shi, E. Cui, H. Zhao, D. Yang, J. Zhu, and S. Dou, “A robust and efficient framework for tubular structure segmentation in chest ct images,” *Technology and Health Care*, vol. 29, pp. 655–665, 2021, 4.

[4] J. H. Noble and B. M. Dawant, “A new approach for tubular structure modeling and segmentation using graph-based techniques,” *Medical image computing and computer-assisted intervention (MICCAI)*, vol. 14, no. Pt 3, pp. 305–312, 2011.

[5] C. Wang, Y. Hayashi, M. Oda, H. Itoh, T. Kitasaka, A. F. Frangi, and K. Mori, “Tubular structure segmentation using spatial fully connected network with radial distance loss for 3d medical images,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2019, pp. 348–356.

[6] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.

[7] S. Javdani, S. Tandon, J. Tang, J. F. O’Brien, and P. Abbeel, “Modeling and perception of deformable one-dimensional objects,” in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1607–1614.

[8] O. Pauly, H. Heibel, and N. Navab, “A machine learning approach for deformable guide-wire tracking in fluoroscopic sequences,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2010, pp. 343–350.

[9] Y. Wang, D. McConachie, and D. Berenson, “Tracking partially-occluded deformable objects while enforcing geometric constraints,” in *2021 International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1–7.

[10] A. Rastegarpanah, R. Howard, and R. Stolkin, “Tracking linear deformable objects using slicing method,” *Robotica*, p. 1–19, 2021.

[11] J. Schulman, A. Lee, J. Ho, and P. Abbeel, “Tracking deformable objects with point clouds,” in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1130–1137.

[12] Y. Lai, J. Poon, G. Paul, H. Han, and T. Matsubara, “Probabilistic pose estimation of deformable linear objects,” in *International Conference on Automation Science and Engineering (CASE)*, 2018, pp. 471–476.

[13] N. Padoy and G. Hager, “Deformable tracking of textured curvilinear objects,” in *2012 23rd British Machine Vision Conference, BMVC 2012*, 2012.

[14] T. Tang, Y. Fan, H.-C. Lin, and M. Tomizuka, “State estimation for deformable objects by point registration and dynamic simulation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2427–2433.

[15] R. Madaan, D. Maturana, and S. Scherer, “Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3487–3494.

[16] A. Zormpas, K. Moirogiorgou, K. Kalaitzakis, G. A. Plokamakis, P. Partsinevelos, G. Giakos, and M. Zervakis, “Power transmission lines inspection using properly equipped unmanned aerial vehicle (uav),” in *2018 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2018, pp. 1–5.

[17] G. Zhou, J. Yuan, I.-L. Yen, and F. Bastani, “Robust real-time uav based power line detection and tracking,” in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 744–748.

[18] Z. Dai, J. Yi, Y. Zhang, B. Zhou, and L. He, “Fast and accurate cable detection using CNN,” *Applied Intelligence*, vol. 50, no. 12, pp. 4688–4707, Dec 2020. [Online]. Available: https://doi.org/10.1007/s10489-020-01746-9

[19] F. Tian, Y. Wang, and L. Zhu, “Power line recognition and tracking method for uavs inspection,” in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 2136–2141.

[20] V. I. Koshelev and D. N. Kozlov, “Wire recognition in image within aerial inspection application,” in *2015 4th Mediterranean Conference on Embedded Computing (MECO)*, 2015, pp. 159–162.

[21] A. Pagnano, M. Höpf, and R. Teti, “A roadmap for automated power line inspection. maintenance and repair,” *Procedia CIRP*, vol. 12, pp. 234–239, 2013, eighth CIRP Conference on Intelligent Computation in Manufacturing Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827113006823

[22] J. Zhang, L. Liu, B. Wang, X. Chen, Q. Wang, and T. Zheng, “High speed automatic power line detection and tracking for a uav-based inspection,” in *2012 International Conference on Industrial Control and Electronics Engineering*, 2012, pp. 266–269.

[23] M. Yan, Y. Zhu, N. Jin, and J. Bohg, “Self-supervised learning of state estimation for manipulating deformable linear objects,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[24] A. Keipour, M. Bandari, and S. Schaal, “Deformable one-dimensional object detection for routing and manipulation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4329–4336, 2022.

[25] O. Merveille, H. Talbot, L. Najman, and N. Passat, “Tubular structure filtering by ranking orientation responses of path operators,” in *Computer Vision (ECCV)*, 2014, pp. 203–218.

[26] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 1998, pp. 130–137.

[27] Y. Wang, X. Wei, F. Liu, J. Chen, Y. Zhou, W. Shen, E. Fishman, and A. Yuille, “Deep distance transform for tubular structure segmentation in CT scans,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3832–3841, 2020.

[28] P. K. Saha, G. Borgefors, and G. Sanniti di Baja, “Chapter 1 - skeletonization and its applications – a review,” in *Skeletonization: Theory, Methods and Applications*, 2017, pp. 3–42.

[29] A. Keipour, M. Eshghi, S. M. Ghadikolaei, N. Mohammadi, and S. Ensafi, “Omnifont Persian OCR system using primitives,” *arXiv:2202.06371*, pp. 1–5, 2013.

[30] P. Maragos and R. Schafer, “Morphological skeleton representation and coding of binary images,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 5, pp. 1228–1244, 1986.

[31] X.-Y. Gong, H. Su, D. Xu, Z.-T. Zhang, F. Shen, and H.-B. Yang, “An overview of contour detection approaches,” *International Journal of Automation and Computing*, vol. 15, no. 6, pp. 656–672, Dec 2018.

[32] A. Keipour, G. A. Pereira, and S. Scherer, “Real-time ellipse detection for robotics applications,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7009–7016, 2021.

[33] S. Suzuki and K. Abe, “Topological structural analysis of digitized binary images by border following,” *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[34] A. Keipour, M. Bandari, and S. Schaal, “Efficient spatial representation and routing of deformable one-dimensional objects for manipulation,” *arXiv:2202.06172*, pp. 1–7, 2022.

[35] A. Keipour, M. Mousaei, A. T. Ashley, and S. Scherer, “Integration of fully-actuated multirotors into real-world applications,” *arXiv preprint arXiv:2011.06666*, 2020.

[36] A. Keipour, “Physical interaction and manipulation of the environment using aerial robots,” Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, May 2022.