POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Robust Shape Tracking of a Deformable Linear Object Manipulated by a Dual-Arm Robot

Laurea Magistrale in Automation and Control Engineering - Ingegneria dell'Automazione

Author: Alessio Russo

Advisor: Prof. Paolo Rocco

Co-advisor: Andrea Monguzzi, Prof. Andrea Maria Zanchettin

Academic year: 2022-2023

1. Introduction

A large number of industrial, household, and medical scenarios involve the manipulation of deformable linear objects (DLOs) such as cables, ropes, wires, and hoses. While the industrial manipulation of rigid objects has been automatized for a long time, the handling of DLOs is usually performed manually, due to their large number of degree of freedom which makes the manipulation of the DLOs a challenging task for robots. Besides finding a solution to planning the robot manipulation, a clear need to determine a way to track the shape of the DLOs while they are manipulated has been identified. Moreover, DLO shape tracking must deal with occlusion, allowing the manipulation to be performed in a constrained environment.

Shape tracking is usually performed using vision sensors: it follows that issues can arise due to lower light conditions, dust, or the presence of a complex background with many DLOs with the same color or similar to the manipulated one.

This thesis purposes the development of a tracking algorithm, which uses only the vision data acquired by a fixed camera and the grippers poses to track online in 3D the shape of different kinds of manipulated DLOs. In addition, the proposed methodology has potential for industrial use, since it is robust to a dynamically changing background, with different kinds of DLOs, also with the same color and shape as the manipulated one. Furthermore, thanks to the acquisition of grippers poses from the robot, the vision algorithm ensures robustness against occlusions, also in case of occluding objects with the same color as the manipulated DLO, an issue not covered in the literature. The methodology is then validated by performing the tracking of different kinds of DLOs manipulated by a real dual-arm robot in different configurations and also in presence of occlusions.

2. State of the Art

Due to the difficulty of tracking the DLOs, different strategies have been presented in the literature. The first is a probabilist method: the Gaussian Mixture Model Expectation-Maximization (GMM-EM) [1]. The GMM-EM method is able to track the deformable object despite noise and outliers but fails in presence of occlusion. For this reason, [2] introduces an EM algorithm that involved physics engines. [3] proposes a novel algorithm called structure preserved registration (SPR) which combined GMM-EM with the Coherent Point Drift (CPD) [4], adding then dynamics simulation for further physical refinement. [5] adjusts the SPR robustness to occlusion, adding a point-cloud recovery inspired by the background

subtraction method in computer vision [6]. All these methods use a physics simulation, which can be computationally expensive, and in addition, are based on a physical model of the DLO and of the environment which can be difficult to model. [7] and [8] propose a method that does not rely on a physics simulator, which ensures tracking while allowing occlusion.

2.1. Thesis contributions

The tracking solution proposed in the literature does not deal with the case of other objects or DLOs also with the same color in the workspace and the case of an occluding object with the same color as the manipulated DLO. Moreover, they fail when the cable tip is occluded [3] or in case of a dynamically changing background [5], e.g. when the manipulated DLO is long enough such that the extremes of it touch the working table and will move during the operation. This limits their use in an industrial scenario.
For this reason, this thesis tries to deal with these limitations by proposing a method to track the shapes of a different class of DLOs, without using probabilistic methods or physics simulation but using only a vision algorithm. This algorithm is able to: (1) Track robustly the DLO also in presence of a dynamically changing background or other objects in the working environment with the same color of manipulated DLO, thanks to the creation of a depth filter.
(2) Track the DLO in case of occlusion, which can be caused by the dual-arm robot occluding the tip of the cable during the manipulation or by the object in the scene. Differently from the presented literature, the occluding object can also be of the same color, thanks to the implementation of the grippers poses in the vision algorithm and the use of a sample video.
(3) Track the shape by performing a 3D geometric estimation of the object.

3. DLO shape tracking algorithm

The algorithm was developed with the aim of tracking different types of DLOs after they are grasped at their ends and raised at a certain high from the working table, where there are scattered different kinds of DLOs. It has 4 main phases: (1) Pre-processing. (2) Segmentation phase. (3) Point-cloud creation. (4) 3D-fitting.

3.1. Pre-processing

In this phase, we define the initial settings for the tracking algorithm, such as the calibration of the camera, and the connection as a client through a socket with the dual-arm robot.

The holes problem generated in the depth image (Figure 1a) can be mitigated using a hole filling filter. In particular, we chose in this phase the fill from left logic, which uses the value from the left neighbor pixel to fill the hole (Figure 1b).

3.2. DLO segmentation

After the pre-processing phase, we can focus on the steps performed at each cycle of the algorithm (Figure 1). First, we need to acquire the color and the depth frame (Figure 1a) from the camera. Then we apply the hole filling on these frames (Figure 1b) and align the depth stream to the color stream, in order to have a one-to-one association between the pixels in the color frame and in pixels in the depth frame, necessary in the point-cloud creation (Section 3.3).

3.2.1. Depth filtering

A challenge of the proposed algorithm is to track the DLO with other DLOs or objects in the background. Considering our assumption that the DLO is manipulated at a certain height from the working table, the idea is to isolate the DLO from what is above and below it. This isolation is made by implementing the depth filter, which returns a new color image based on the depth one: this is possible thanks to the previous alignment of the depth frame with the color frame. The depth filter puts the new color image pixels between the upper and lower depth thresholds (obtained considering the grippers positions) equal to the pixel of the acquired color frame, otherwise, these pixels will be colored in orange (Figure 1c).

3.2.2. Color mask

Once the DLO is partially isolated by the depth filter, it is possible to apply a color mask. The color mask searches in each frame the pixels with a certain specified color, putting in white all the pixels with this feature and in black the others (Figure 1d). It is easy to understand the importance of applying first the depth filter and then the color mask: if we didn't use the depth filter, the color mask would segment all objects of the same color of the manipulated DLO causing a wrong segmentation of the DLO.

3.2.3. Contour extraction

Once the output of the color mask is obtained, we can apply a contour extraction algorithm to it (Figure 1e). The contour extraction algorithm finds all the contours in the masked image, saving all points

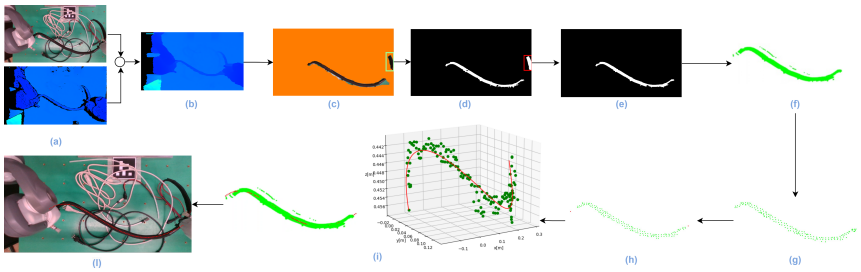


Figure 1: (a) Acquisition of the color and depth frame. (b) Application of the hole filling filter. (c) Depth filtering, in the green rectangle the DLO part after the right gripper. (d) Color mask, in the red rectangle the DLO part after the right gripper. (e) Contour extraction, which filters out the DLO part after the right gripper. (f) Point-cloud construction. (g) Point-cloud down-sampling and outliers removal. (h) Acquisition and addition of the grippers poses (red points) to the point-cloud. (i) 3D-fitting, in red the tracked shape. (l) DLO tracked shape (in red) on the color frame.

along the boundaries, with the same color and intensity. It then records all the discovered contours' perimeters, identifying the longest one. Finally, only the pixels in the perimeters that are bigger or equal to $\frac{1}{5}$ of the longest perimeter are colored in white. Then the image is transformed into grayscale in this way the proposed online algorithm becomes more computationally efficient.
The contour extraction step ensures the segmentation only of the DLO between the two grippers, extending the use of this tracking methodology to different kinds of DLOs without any limit on their lengths. In Figure 1a, the piece of the DLO after the right gripper has the same depth value as the part between the grippers. For this reason, the depth filter isolates also the DLO piece after the right gripper (in green in Figure 1c). Therefore, if we only use the color mask it would segment both parts (Figure 1d), and this will affect the final tracked shape. Instead, the contour extraction (Figure 1e) ensures the segmentation only of the DLO between the gripper, since the piece after the right gripper has a perimeter too small with respect to the perimeter of the DLO between the gripper. In addition, it ensures the construction of the point-cloud (further details in Section 3.3) with a lower number of outliers in case of occlusion, due to the fact they are filtered by the perimeter threshold.

3.3. Point-cloud creation

After the execution of the contour extraction phase, the DLO pixels are in white, so the pixels coordinates can be extracted from it. Knowing that our color frame and depth frame are aligned, each pixel has associated also the depth information. Finally, each extracted pixel in the image plane is projected in the space, constructing the point-cloud (Figure 1f). Then, a voxel downsample [9] of the point-cloud is necessary in order to make the fitting (Section 3.4) computationally efficient. The outlier can be removed by applying a statistical removal [10], that removes the points that are further away from their neighbor compared to the average distance of the points in the point-cloud (Figure 1g).

3.3.1. Acquisition of gripper pose

The application of the voxel downsample and the removal of the outliers ensures the right balance between the number of points and meaningful points in the point-cloud for all the DLO considered. However, there can be cases where these points are not enough for good tracking, due to: occlusions, the presence of particular light conditions, or particular characteristics of the DLO. One of the ways to resolve these problems is to acquire during the tracking the grippers poses from the robot. This is possible through a socket communication between the tracking algorithm and the dual-arm robot. In this way, during the tracking, two points (one for each gripper), are added to the point-cloud. Note that they are not influenced by the previous problems. These points are added to the filtered point-cloud (Figure 1h).

3.4. 3D-Fitting

After the previous phase, we obtained a point-cloud that represents the DLO in space, defined as (X,Y,Z), with $X = [X_1, X_2 ... X_P]^T$, $Y = [Y_1, Y_2 ... Y_P]^T$, $Z = [Z_1, Z_2 ... Z_P]^T$. These points are exploited to construct a geometric estimation of the DLO shape, by applying a 3D-fitting. The 3D-fitting is applied by fixing the x-axis and then fits the other two sets of coordinates with respect to the x-axis. In this way, the fitting problem becomes simpler and more computationally efficient. Accordingly, the 3D fitting problem is decomposed into two 2D Lasso regression problems: one in the x-y plane and another for the x-z plane.

Formalized as:

$$\begin{aligned} \min_{c_y} \quad & \frac{1}{2P}||Y - Vc_y||_2^2 + \alpha||c_y||_1 \\ \min_{c_z} \quad & \frac{1}{2P}||Z - Vc_z||_2^2 + \alpha||c_z||_1 \end{aligned} \tag{1}$$

where V is the Vandemorde matrix filled with X and with maximum degree of four, being enough expressive to describe the shape that a manipulated DLO can assume. α is used to weight regularization terms. $c_y, c_z \in \mathbb{R}^5$ are the vectors of the polynomial coefficient. For each acquired frame, the algorithm decided what is the best polynomial, from grade zero to four, that describes the DLO shape in the x-y and x-z planes.
Solving the Lasso problems Equation (1), two polynomials are obtained, one for the x-y plane and another for the x-z plane:

$$\begin{aligned} P_{x-y} &= c_{y4}X_{new}^4 + c_{y3}X_{new}^3 + c_{y2}X_{new}^2 + c_{y1}X_{new} + c_{y0} \\ P_{x-z} &= c_{z4}X_{new}^4 + c_{z3}X_{new}^3 + c_{z2}X_{new}^2 + c_{z1}X_{new} + c_{z0} \end{aligned}$$

with $X_{new} \in \mathbb{R}^{100}$ that goes from the minimum to the maximum of X data. They describe the final tracked shape in 3D (Figure 1i), that can be then reported in the image plane and can be plotted on the color frame (Figure 1l).

4. Robustness to occlusion

The proposed methodology ensures robustness to occlusions since the DLO segmentation phase (Section 3.2) guarantees complete isolation and extraction of the DLO pixels. Then, once the DLO pixels are reported in the 3D, the gripper points are also added, ensuring more robustness in the shape tracked by the Lasso regression. Figure 2c, shows an example: the robot left arm occluded the left end of the DLO, but thanks to the acquisition of the grippers poses, the occluded part can be estimated. In addition, it shows an example of the tracking result in case of a hand of the operator occludes the DLO touching it. Note that the depth filter (Figure 2a) isolates also the hand since it touches the DLO, but the hand has different color with respect to the DLO, so the color mask will segment only the DLO ensuring the tracking of the shape.

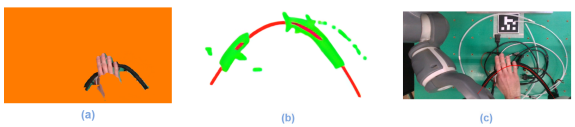


Figure 2: (a) Depth filter output. (b) DLO point-cloud with tracked shape. (c) Color frame with tracked shape in red.

4.1. Sensitivity to the occluding object color

One of the limitations of the presented methodology, as in the ones presented in the literature, is the use of a color mask in order to segment the DLO. This means that in the case the occluding object has the same color as the manipulated DLO, the color mask would segment also the object causing a wrong shape estimation. Note that the methods proposed in the literature consider only occluding objects with different colors with respect to the tracked one.
In the proposed methodology this limitation is mitigated, thanks to the application of the depth filter (Section 3.2.1) before the color mask. Indeed if the occlusion happens above the manipulated DLO of approximately 4 cm, the depth filter will cut off the occluding object (Figure 3a), ensuring good tracking results (Figure 3b and Figure 3c). Meanwhile, if the black object is on the DLO touching it, the depth filter isolate also the black object (Figure 3d). Consequently, the color mask and the contour extraction acquires also the black object pixels, so they are considered in the fitting causing a wrong-tracked DLO's shape (Figure 3e and Figure 3f). This limit on the upper threshold of the depth filter is linked to infinite degrees of freedom of the DLOs, which can also deform upward during the manipulation. This deformation depends on the rigidity of the DLO, but the proposed algorithm tracks the DLOs without the use of rigidity information. So choosing an upper threshold very strict can deletes important parts of the DLO.

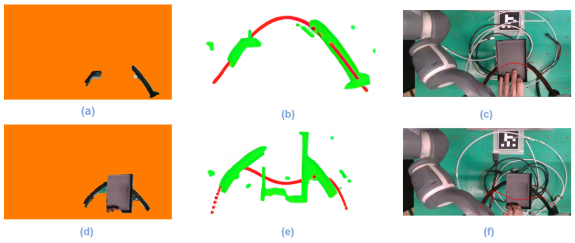


Figure 3: (a) Depth filter output in case of the black object 4 cm above the DLO. (b) DLO point-cloud with tracked shape, in case of the black object 4 cm above the DLO. (c) Color frame with tracked shape in red, in case of the black object 4 cm above the DLO. (d) Depth filter output in case of the black object touching the DLO. (e) DLO point-cloud with tracked shape, in case of the black object touching the DLO. (f) Color frame with tracked shape in red, in case of the black object touching the DLO.

This thesis is developed to ensure tracking of a manipulated DLO in an industrial scenario where the same manipulation is repeated multiple times, it is then reasonable to assume that there is a sample video of

the manipulation without any occlusion, in order to demonstrate the manipulation and the final desired shape. The mentioned problem can be completely solved by running in parallel to the online acquired video from the camera, also the sample video. Applying on both videos all the pre-processing (Section 3.1) and segmentation (Section 3.2) steps, but adding the AND operation between the outputs of the contour extraction phase on the online video and on the sample video. In this way, before the construction of the point-cloud, all the object points are filtered out (Figure 4e).

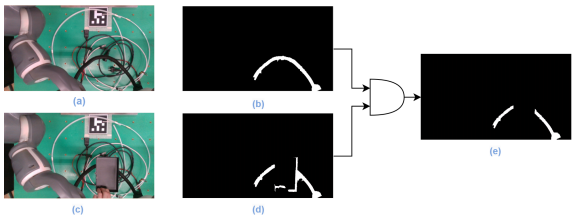
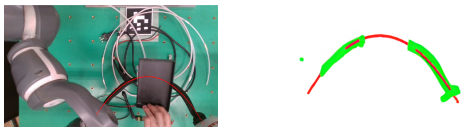


Figure 4: (a) Color frame of the sample video without the black occluding object. (b) Result of the contour extraction on the sample video without the black occluding object. (c) Color frame of the online video with black occluding object. (d) Result of the contour extraction on the online video with black occluding object. (e) The AND between (b) and (d), filter out the object points.

Figure 5 shows the tracking result during a dynamic occlusion, once the object points are filtered out by the AND operation, the tracking methodology is able to estimate DLO, differently from Figure 3.



(a) (b)

Figure 5: (a) DLO color frame with tracked points (in red). (b) DLO point-cloud (in green) with tracked points (in red).

Despite this method is based on the application of the DLO segmentation step two times, one for the online video and one for the sample video, there is no degradation of the computational time since all the operations are made in grayscale. In fact, the mean computational time is 0.15 s, which increases only by 0.03 s with respect the case without occlusion (DLO1 in Table 1).

5. Experimental analysis

A series of experimental tests were performed to validate the proposed tracking methodology in an industrial scenario, where the DLO is manipulated at a certain high from the working table with different kinds of DLOs are scattered on it. In particular, the performed tests involved the manipulation of 8 DLOs (which differ in color, rigidity and dimension) in different shapes, from the less complex one, as the linear shape, to the more complex one, as the sinusoidal shape. Due to space limitations, we do not describe here each experiment, but instead, we present the most relevant ones. The experimental setup is performed through an ABB YuMi Robot, a Intel RealSense D435i camera, and the tracking algorithm is implemented in Python on an Intel i7-12650H processor with 16 GB RAM. For each experiment and for each DLO, the averaged mean squared error (AMSE) in the x-y and x-z planes was analyzed:

$$AMSE_{x-y} = \frac{1}{n_{frames}} \sum_{j=1}^{n_{frames}-1} \sum_{i=1}^{P-1} (Y_i - Y\_true_i)^2$$

$$AMSE_{x-z} = \frac{1}{n_{frames}} \sum_{j=1}^{n_{frames}-1} \sum_{i=1}^{P-1} (Z_i - Z\_true_i)^2$$

with

$$Y\_true = c_{y4}X^4 + c_{y3}X^3 + c_{y2}X^2 + c_{y1}X + c_{y0}$$
$$Z\_true = c_{z4}X^4 + c_{z3}X^3 + c_{z2}X^2 + c_{z1}X + c_{z0}$$

where X, Y, and Z are the acquired points coordinates from point-cloud. P is the number of points, c_y and c_z are the results of the Lasso regression (Equation (1)).

5.1. Test1: Quadratic function shape

This shape is compliant to different kinds of DLO, from the less rigid to the more rigid one. (Figure 6) show the tracking results with and without occlusion for different DLOs. Figure 2 and Figure 6a and show the tracking on a black branched DLO (DLO1) with a diameter of 1.55 m and low rigidity. Note that due to lower rigidity, the DLO1 is deformed downward by 2.5 cm. Figure 6b and Figure 6c shows the tracking results on a hose of a motorbike braking system with a diameter of 0.75 cm (DLO2), characterized by the highest rigidity in the tested DLOs, and by a reflective material that makes difficult to acquire its contour by the color mask. Figure 6d and Figure 6e show the tracking results on a white ethernet cable with a diameter of 0.6 cm (DLO3). Figure 6f and Figure 6g show the tracking results on a hose for compressor air with a diameter of 0.6 cm, which is translucent

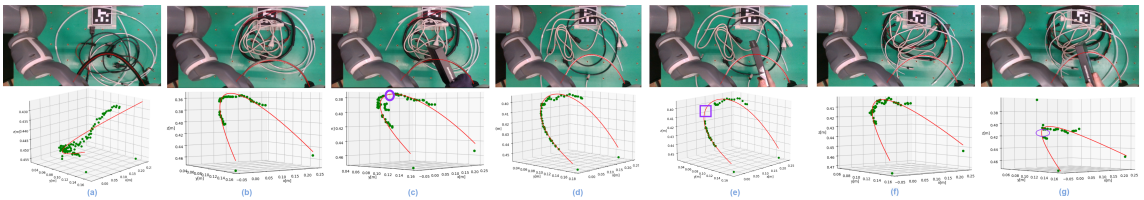


Figure 6: Quadratic function shape tested on different DLOs with and without occlusion. The top row shows the color frame and the reprojection of the estimated shape (in red color); and the bottom row shows the 3D fitting plot, in green the point-cloud, in red the fitted shape and in purple the occlusion. (a) Black branched DLO (DLO1). (c-b) Hose of a motorbike braking system (DLO2). (d-e) White ethernet cable (DLO3). (f-g) Hose for compressed air, which is translucent (DLO4).

and creates some issues in the detection of its depth values. In all 3 cases due to high rigidity, forcing them to reach a final shape parallel to the working table generates a deformation upward of more than 5 cm in the DLO3, 6 cm in the DLO4, and 10 cm in the case of the DLO2. Although, the particular features of the DLOs and without any information about their rigidity, the tracking algorithm ensures satisfactory results in terms of mean computational time (MCT, Table 1) and AMSE (Table 2).

MCT for each tracked frame		
	without occlusion	**with occlusion**
DLO1	0.12 s	0.116 s
DLO2	0.085 s	0.082 s
DLO3	0.081 s	0.076 s
DLO4	0.068 s	0.066 s

Table 1: Mean computational time for each tracked frame with occlusion and without it.

$AMSE_{x-y}$		
	without occlusion	**with occlusion**
DLO1	0.609 cm^2	0.56 cm^2
DLO2	0.045 cm^2	0.15 cm^2
DLO3	0.032 cm^2	0.035 cm^2
DLO4	0.045 cm^2	0.089 cm^2

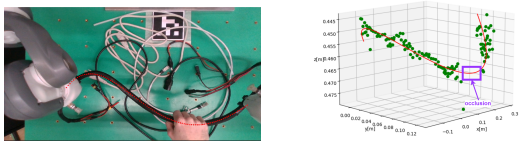
$AMSE_{x-z}$		
	without occlusion	**with occlusion**
DLO1	0.282 cm^2	0.284 cm^2
DLO2	1.5 cm^2	2 cm^2
DLO3	0.091 cm^2	0.23 cm^2
DLO4	3.8 cm^2	3.3 cm^2

Table 2: $AMSE_{x-y}$ and $AMSE_{x-z}$.

5.2. Test2: 2D-Sinusoidal shape

Figure 1 shows this shape on the DLO1, in particular, Figure 7 shows a test where the DLO is occluded by a hand that holds it and slides along it. As can be noticed from Figure 1i due to lower rigidity the DLO is deformed downward, in particular in Figure 7: inevitably the operator's hand interacting with the DLO generates a deformation downward of 2 cm. The methodology, without any information about the rigidity of the DLO but using only the vision data correctly track these deformations.

The mean computational time for each tracked frame without occlusion is 0.104 s, while in the case of occlusion is 0.103 s. Then the $AMSE_{x-y}$ without occlusion is 0.389 cm^2, while in the case of occlusion is 0.427 cm^2. Finally the $AMSE_{x-z}$ without occlusion is 0.054 cm^2, while in the case of occlusion is 0.077 cm^2.

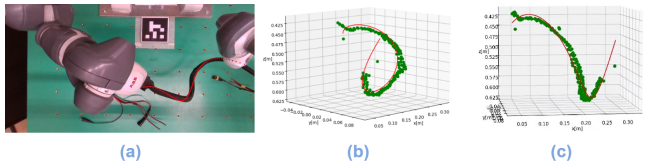


(a) (b)

Figure 7: (a) DLO color frame with tracked shape (in red) in case of a hand occlusion. (b) DLO 3D-fitting plot, with occlusion highlighted in purple.

5.3. Test 3: 3D-Sinusoidal shape

The last tested shape is the 3D sinusoidal shape, similar to the one of Section 5.2 but with a huge variation along the z-axis. Figure 8 shows the DLO1 result. As can be noticed by Figure 8, $\tilde{Z} = 20$ cm, where $\tilde{Z} = Z_{max} - Z_{min}$. Nearly 10 times the variation that we had in the previous sinusoidal shape (Figure 1i). The mean computational time for each tracked frame is 0.11 s, while the $AMSE_{x-y}$ is 0.5 cm^2, while the $AMSE_{x-z}$ is 0.4 cm^2.



(a) (b) (c)

Figure 8: DLO1 3D sinusoidal shape manipulation. (a) DLO1 color frame with tracked shape (in red). (b) DLO1 3D-fitting plot. (c) DLO1 3D-fitting plot.

6. Conclusions

The proposed thesis aims the development of tracking algorithm to track robustly the shape of a deformable linear object manipulated by a dual-arm robot.
The experimental results demonstrate our method can guarantee a computationally efficient and smooth shape of DLOs in unoccluded or occluded cases, even in cases of other DLO in the workspace or occluding objects with the same color as the manipulated DLO. Those behaviours are not affected by the material composing the cables, since the algorithm is constructed to be independent of the treated DLO.
The methodology can be improved by combining the proposed online tracking strategy with an offline planner, which can be used as a reference. Moreover, the tracking algorithm can be used to realize a visual servoing strategy.

References

[1] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No.PR00737)*, pp. 190–197, 2000.

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

[3] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, vol. 41, pp. 599 – 614, 2019.

[4] A. Myronenko and Song, "Point-set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 2262–2275, 2010.

[5] S. Jin, C. Wang, X. Zhu, T. Tang, and M. Tomizuka, "Real-time state estimation of deformable objects with dynamical simulation," p. 11879–11883., 11 2020.

[6] S. R. Balaji and S. Karthikeyan, "A survey on moving object tracking using image processing," in *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, pp. 469–474, 2017.

[7] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6443–6450, 2019.

[8] Y. Wang, D. McConachie, and D. Berenson, "Tracking partially-occluded deformable objects while enforcing geometric constraints," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 14199–14205, 2021.

[9] Open3D, "Voxel down sample." `http://www.open3d.org/docs/0.6.0/python_api/open3d.geometry.voxel_down_sample.html`.

[10] Open3D, "Statistical outlier removal." `http://www.open3d.org/docs/0.6.0/python_api/open3d.geometry.statistical_outlier_removal.html`.