Deformable Linear Objects 3D Shape Estimation and Tracking From Multiple 2D Views

Deformable Linear Objects 3D Shape Estimation and Tracking From Multiple 2D Views

Alessio Caporali, Kevin Galassi, and Gianluca Palli, *Senior Member, IEEE*

***Abstract*—This letter presents *DLO3DS*, an approach for the 3D shapes estimation and tracking of Deformable Linear Objects (DLOs) such as cables, wires or plastic hoses, using a cheap and compact 2D vision sensor mounted on the robot end-effector. *DLO3DS* can be applied in all those scenarios in which the perception and manipulation of DLO-like structures are needed, such as in the case of switchgear cabling, wiring harness manufacturing and assembly in the automotive and aerospace industries, or production of hoses for medical applications. The developed procedure is based on a pipeline that first processes the images coming from the 2D camera extracting key topological points along the DLOs. These points are then used to model each DLO with a B-spline curve. Finally, the set of splines obtained from all the images is matched by exploiting a multi-view stereo-based algorithm. *DLO3DS* is validated both on a real scenario and on simulated data obtained by exploiting a rendering engine for photo-realistic images. In this way, reliable ground-truth data are retrieved and utilized for assessing the estimation error achievable by *DLO3DS*, which on the employed *test set* is characterized by a mean reconstruction error of 0.82 mm.**

***Index Terms*—Deformable linear objects estimation, shape detection, robotic vision, reconstruction, 3D.**

I. Introduction

NOWADAYS the request for the automation of processes involving cables, wires, hoses, wiring harnesses, and in general Deformable Linear Objects (DLOs), is relevant in many industrial manufacturing areas. As an example, in the automotive and aerospace sectors, the assembly of the cabling systems is actually an expensive process almost completely based on human work [1]. On the other hand, automating aspects concerning the manipulation of DLOs is not easy. In fact, manufacturing processes involving DLOs pose serious problems at both the manipulation and perception levels [2] since their intrinsic deformability makes modeling their behavior during the manipulation complex. Moreover, if connectors or other rigid parts attached to them are missing, the DLOs lack of features and significant textures make their detection with vision systems challenging even with new state-of-the-art learning-based methods [3], [4].

Manuscript received 20 January 2023; accepted 28 April 2023. Date of publication 8 May 2023; date of current version 17 May 2023. This letter was recommended for publication by Associate Editor Z. Min and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported by the European Commission's Horizon 2020 Framework Programme with the project REMODEL - Robotic technologies for the manipulation of complex deformable linear objects - under grant agreement No 870133. *(Corresponding author: Alessio Caporali.)*

The authors are with the DEI - Department of Electrical, Electronic and Information Engineering, University of Bologna, 40136 Bologna, Italy (e-mail: alessio.caporali2@unibo.it; kevin.galassi2@unibo.it; gianluca.palli@unibo.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3273518, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3273518

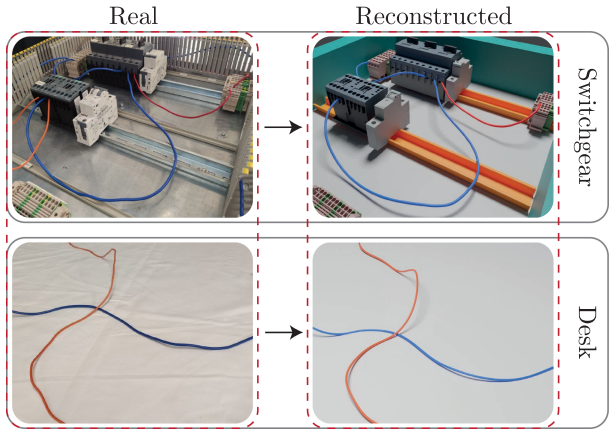

Fig. 1. Showcase of *DLO3DS* capabilities in reconstructing the shapes of DLOs in different scenarios.

In this letter, we analyze the problem of the accurate estimation of the DLOs 3D shape. In this regard, general purpose *consumer* 3D cameras like Intel RealSense or CamBoard pico flexx fails in perceiving thin objects like DLOs [5]. This problem is shared across all *consumer* devices irrespective of the specific 3D depth technologies. The only category of 3D active cameras that can reliably detect the shape of very thin cylindrically shaped objects like DLOs with a diameter as low as 2-3 mm is the high-end one, consisting of devices like Zivid One+/Two and Photoneo MotionCam3D or short-range laser scanners [5]. In fact, these devices can reach sub-millimeter depth accuracy, but, on the other hand, they show several limitations in terms of pricing, bulkiness, and working constraints. Thus, they are usually placed at a fixed position and not at the end-effector level, increasing the risk of occlusions and reducing the flexibility of the application. If semi-transparent materials are taken into account, such as in the case of medical hoses manufacturing, even high-end 3D sensors are not able to correctly detect those materials because of transparency, refraction and internal reflections [5].

In contrast, 2D cameras arranged in a stereo (or multi-view stereo) setup could potentially be more effective in detecting thin DLOs. However, these passive 3D devices have limitations in terms of baseline (which is fixed and optimized for distant objects) and usually struggle in case of changes in lights and non-textured areas. DLOs, having small dimensions and lacking relevant textures, represent a difficult object to tackle for passive stereo cameras.

To address the drawbacks of both 3D active and passive cameras, we decided to deploy a single 2D camera mounted on

a robotic arm. Utilizing just a 2D camera brings many beneficial effects: these cameras are usually cheaper than 3D devices, more compact and lighter, they have a wide range of resolutions, and the field of view and working distance can be easily adapted to the specific scenario. In addition, placing the 2D sensor on the robotic arm allows for exploiting the high repeatability and accuracy of the latter to avoid occlusions while, at the same time, enabling immense flexibility in terms of baselines and distances from the target.

In this letter, a method to infer the 3D shape of DLOs in static scenes by exploiting DLO instances [4] extracted from multiple images is introduced. For the sake of brevity, the proposed method is referred to as *DLO3DS* in the following. *DLO3DS* exploits a multi-view stereo-based approach to reconstruct the 3D DLO shape from multiple images taken at known viewpoints without any prior knowledge of the DLOs or the surrounding scene, independently from the background. *DLO3DS* extends the preliminary results obtained in [6]. The DLO instances, after being modeled as B-spline curves, are matched by exploiting a triangulation-based method. *DLO3DS* provides reliable results where standard stereo-matching algorithms [7] fail due to the peculiar characteristics of DLOs previously discussed. Finally, the availability of the robotic arm is exploited by optimizing at run-time the baseline and distance from the target, thus reducing, even more, the estimation error. In Fig. 1 the capabilities of *DLO3DS* in two different scenarios are shown.

The contributions of this letter can be summarized as:

- B-splines modeling of the DLO shape in image coordinates and reliable estimation of the shape of a target DLO in the 3D space;
- Optimization at run-time of baseline and distance from the target exploiting the robotic arm and camera on the end-effector;
- Extensive analysis of the accuracy and error characteristics of the 3D estimations with comparisons against several stereo-based baseline methods;

The *DLO3DS* source code is available at https://github.com/lar-unibo/DLO3DS.

II. Related Works

A. 3D Reconstruction From Multiple Views

The 3D shape reconstruction of objects from 2D images is a complex and extensively analyzed problem in computer vision. In this letter, we develop a multi-view stereo approach for the 3D estimation of DLOs. The goal of multi-view stereo is to reconstruct a complete 3D object model from a collection of images taken from known camera viewpoints [8]. In this section, we review the closest contributions and methods dealing with stereopsis. In order to achieve high accuracy in the 3D reconstruction, we should work on both the *disparity* error and the *geometric* error [9]. The first is related to correspondence algorithms while the latter is to physical parameters like baseline and distance from the objects. In the context of correspondence algorithms, stereo approaches are usually classified between local and global methods [10]. The latter are usually slower but more effective than the first in the case of non-textured areas. Among the many existing approaches, Semi-Global Matching (SGM) [7] is the most widely used approach due to its balance between quality, efficiency and scalability. However, its limitations in the case of non-textured areas are well-known [11] and several works have tried to address its weaknesses, such as time execution with a GPU implementation [12]. With the rise of deep learning, several approaches have been proposed for the computation of correspondence by employing SGM with, for instance, learned parameters [13], learned matching cost [14], a complete end-to-end learning approach [15]. Learning methods could potentially solve several challenges of traditional stereo algorithms, although the problem of dataset generation and model deployment in the real world still remains to be evaluated.

Concerning the geometric error, it can be not possible to adjust the baseline in case of a fixed stereo setup, as well as with commercial solutions. Thus, only the distance of operation (and possibly the resolution) can be modified. Instead, some works exploit either multiple 2D cameras mounted with different baselines to combine the advantages of short and wide baseline systems [16], whereas others employ a single 2D camera and a robot to emulate a multi-baseline system [17]. *DLO3DS* tackles both the *disparity* and *geometric* errors. The first is addressed by the reliable processing of the 2D images and the matching of splines. The latter is by exploiting the robotic arm optimizing at run-time the baseline and the distance from the target.

B. 3D Perception in Robotics

Common sensors that can be found in robotics are from the RealSense and PrimeSense families [18] or more expensive ones such as the Ensenso 3D camera [19]. Very recently, new highly accurate 3D active sensors were made available, like the ones from Zivid or Photoneo. From [5] emerged their suitability for reconstructing very thin and small objects. Alternatively, Linear Laser Scanners mounted in an eye-in-hand configuration can be considered as well, in case an even higher reconstruction accuracy is sought [20]. Despite the abundance of sensors and 3D technologies, there still exist some limitations in case a specific application, like the one presented in this letter, requires the utilization of the device very close to the target while satisfying space constraints as well [5]. *DLO3DS* solves these limitations by empowering a compact 2D camera mounted on the robot end-effector.

C. DLOs Detection and Segmentation

Concerning the detection and modeling of DLO in images, several approaches have been described in the literature [2]. The semantic segmentation of DLOs, specifically electric wires, via learning-based methods has been attempted in [21] where a dataset is made publicly available. Similarly, a weakly supervised dataset generation approach combining synthetic and real images of DLOs has been proposed in [22]. Simpler methods to segment DLOs in images are based on markers [1], background color removal [23], [24], [25], Frangi filter [26], Ridge filter [27] or ELSD algorithm [28].

More complex approaches are [3], [4], [29]. In particular, these methods allow obtaining the individual instances composing the scene. In *Ariadne* [29], the individual DLOs are segmented from complex backgrounds starting from their endpoints, which are detected by a CNN. *Ariadne+* [3] improves *Ariadne* by building the graph representation directly from the segmentation mask avoiding the CNN step. In combination with a more efficient paths discovery algorithm, better accuracy and

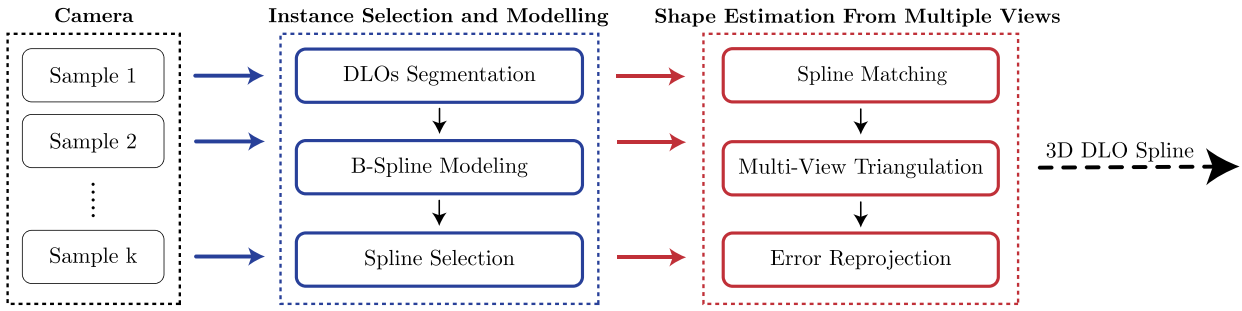

Fig. 2. 3D shape estimation pipeline of *DLO3DS*. The set of k image samples is processed by an instance selection algorithm for extracting the DLOs from each individual image. A B-spline model is computed for each detected instance and a single 2D target spline is selected from each image sample. The set of selected B-splines is matched prior to performing the triangulation procedure, obtaining a 3D spline describing the DLO shape in world coordinates. Data flow: the blue arrow denotes the image; the red arrow denotes the 2D spline.

a noticeable speedup are achieved. Recently, a 20 Frames-Per-Second (FPS) capable approach named *FASTDLO* [4] was proposed, further boosting accuracy and throughput. These methods were then exploited in bigger frameworks in order, for instance, to combine the sensing of DLO also from tactile sensors [30]. In [6], instead, some preliminary results about the shape estimation of DLOs are provided. Notice that *DLO3DS* contains several improvements with respect to [6], like being able to deal with multiple DLOs in the scene and to track a target DLO shape. In addition, extensive experimental validations and comparisons are provided.

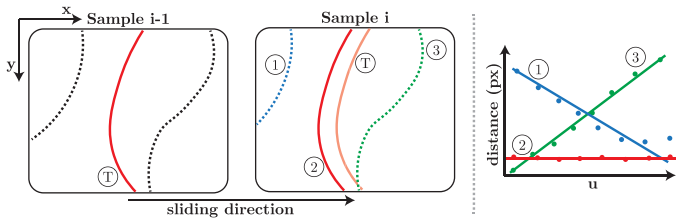

Fig. 3. Target spline selection approach based on distance computation. The symbol u denotes the spline free parameter.

III. Instance Selection and Modeling

This section reports the details about the processing and estimation of a spline for each captured image. The estimated spline is employed both for computing the 3D shape of the DLO but also for aligning the camera with the target DLO main direction through Principal Component Analysis (PCA). Indeed, in order to increase the portion of the same DLO visible in every sample, it is assumed to have the camera oriented along the DLO main axis and to record the samples by sliding orthogonally to it, see Fig. 3 for an example of the sliding direction with respect to the DLOs orientation. In Section III-A, the extraction of DLO instances from each image and their modeling via B-splines curves is presented. Section III-B discusses the selection of the target spline among the set of detected ones extracted from a captured sample.

A. DLOs Segmentation and B-Spline Modeling

DLO3DS exploits existing approaches for segmenting the DLOs from an image. In this work, the learning-based algorithm named *FASTDLO* [4] is employed, taking as input the RGB image of the scene and providing as output both an instance mask, where each DLO is denoted with a unique color identifying the assigned ID, and a sequence of 2D coordinates in the image plane for each detected DLO. A cubic B-spline is fitted to these coordinate points obtaining a continuous representation of the considered DLO. The considered spline is addressed as $q(u)$, where $u \in [0,\ 1]$ is the free parameter, i.e. the normalized position along the spline neutral axis. The computed curve is then discretized into a fixed number n_s of points. The utilization of a learning framework in [4] allows to intrinsically deal with changes in lights and textureless areas, partially solving the limitations discussed in Section I. However, other image processing pipelines can be employed for increased robustness and depending on the application scenario.

B. Spline Selection

The spline selection is performed in case an image contains multiple DLOs. Indeed, all the instances extracted from an image are modeled in Section III-A. However, in the following, a single DLO spline per image is expected. Thus, a regression-based distance approach is employed for retrieving the target DLO in sample i, based on sample $i-1$, with $i = 2, \ldots, k$. In particular, the point-to-point distance between the target spline T of a sample $i-1$ and each newly detected spline of a sample i is computed, as shown in Fig. 3. Then, a line is regressed for each distance curve and the spline associated with the smaller slope line is selected. Indeed, due to the orthogonal sliding direction, the same portion of DLO is assumed to be visible in each sample, thus an overall constant distance between the two curves given by the motion of the baseline step is expected.

IV. Shape Estimation From Multiple Views

This section details how the different k splines are matched and exploited for obtaining the final 3D shape of a given DLO, see Fig. 2. In Section IV-A the matching of the splines is discussed, while in Section IV-B the triangulation approach is detailed. In Section IV-C the possibility of employing the reprojection error for evaluating the quality of the estimation is presented. In Section IV-D, the optimization of the baseline and distance from the target is described. Finally, in Section IV-E, the applicability of *DLO3DS* in a DLO tracking framework is analyzed.

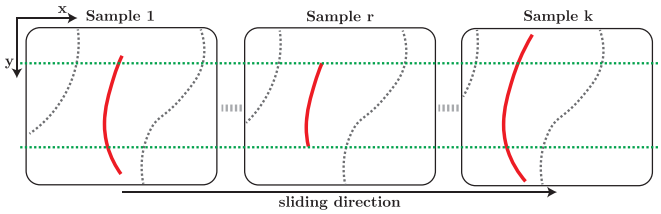

Fig. 4. Scaling process. The shortest spline is selected as the reference and all the others are scaled to match the same DLO portion as closely as possible.

A. Splines Matching

A spline $q_i(u)$ can be sampled by defining a suitable vector u of n_s equally-spaced free parameter values in the interval $[0,\ 1]$. Thus, n_s 2D pixel points along the DLO for the i-th view are retrieved.

Let's denote with $p_{ij} = [p_{x_{ij}}\ p_{y_{ij}}]^T$ the j-th spline sample on the i-th image plane, with $i = 1, \ldots, k$ and $j = 1, \ldots, n_s$. To assess the accurate 3D location of a generic point seen from multiple images at pixel coordinates p_{ij}, we need to compute precisely the corresponding points p_{ij}. For this purpose, we exploit both the constraints embedded in the case of a normal stereo setup and the availability of the splines modeling the DLO.

The first step consists of sampling all the splines over the same DLO section by defining suitable vectors u_i, one for each spline. The length l_i of each spline is measured by summing the distance in pixels among adjacent points. Then, the index r of the shortest spline is taken as a reference

$$r = \text{argmin}_i\{l_i : i \in 1, \ldots, k\}$$

Thus, the splines are re-sampled according to the redefined vectors of free parameters

$$u_i \leftarrow u_i \frac{l_i}{l_r} + \frac{d(q_i(0), q_r(0))}{l_i}$$

where the function $d(\cdot, \cdot)$ provides the distance in pixels between two points. As a consequence, the spline samples $q_i(u_i)$ provide a coarse matching across the different views.

The n_s spline samples of the shortest spline need to be precisely matched in all the other splines $q_i(u)$, $i = 1 \ldots k \setminus r$. In this regard, the corresponding j-th point on the i-th image plane p_{ij} is searched along the row coordinate of p_{rj}, empowering the basic constraints of epipolar lines in case of a normal stereo rig, as the intersection point with the spline $q_i(u)$. In the eventuality of multiple matches between the spline curve and the epipolar line, a smoothness constraint is also employed enforcing the most consistent point based on the past matches.

The aforementioned procedure is depicted in Fig. 4. The spline samples p_{ij} are then used to compute the DLO 3D shape as detailed in Section IV-B.

B. Multi-View Triangulation

For the sake of simplicity, the discussion is focused first on just one target point, i.e. $j = 1$. Let us consider the case in which a single unknown point p in the Cartesian space expressed with respect to the world reference frame is observed by the camera mounted on the robot from multiple points of view. Provided that the camera frame with respect to the world frame at the i-th points of view is

$${}^wT_{c_i} = \begin{bmatrix} {}^wR_{c_i} & {}^wt_{c_i} \\ 0\,0\,0 & 1 \end{bmatrix}$$

where ${}^wR_{c_i}$ is the rotation matrix and ${}^wt_{c_i}$ is the position of the camera frame origin in world coordinates obtained from the kinematics of the robot and the extrinsic parameter of the camera calibration. It is assumed that the point p is seen in the image related to the i-th points of view at $p_i = [p_{x_i}\ p_{y_i}]^T$, being p_{x_i} and p_{y_i} the point pixel coordinates in the image.

A so-called unit ray v_i passing through the image reference frame origin and p can be expressed in the image frame considering the pixel coordinates p_i and the camera focal distance f

$$v_i' = \begin{bmatrix} p_{x_i} - c_x \\ p_{y_i} - c_y \\ f \end{bmatrix}, \quad v_i = \frac{v_i'}{\|v_i'\|}$$

where c_x and c_y are the pixel coordinates of the image center (assuming the camera frame is centered with respect to the image). Then, v_i can be expressed in the world frame by

$${}^wv_i = {}^wT_{c_i} v_i$$

Provided that k distinguished points of view are available, the estimation $\tilde{p}$ of the unknown point p can be obtained by looking for the point having the minimum distance from all the rays. By defining the symmetric V_i matrix

$$V_i = I - {}^wv_i\,{}^wv_i^T \tag{1}$$

providing the semi-norm on the ray distance, the point location estimate $\tilde{x}$ is provided by the nearest point search algorithm, i.e.

$$\tilde{p} = \left(\sum_{i=1}^{k} V_i\right)^{-1} \left(\sum_{i=1}^{k} V_i\,{}^wt_{c_i}\right)$$

The aforementioned algorithm is thus applied to estimate the DLO segment employing as input the spline samples $p_{ij} = p_i(u_j), j = 1, \ldots, n_s, i = 1, \ldots, k$. The vector of control points $q_v = [q_1\ \cdots\ q_{n_s}]^T$ of the 3D spline $q(u)$ that optimally approximated the set of point estimates p_{ij} can be defined as

$$q_v = B^{\#}\tilde{x}_v$$

where $^{\#}$ represents the matrix pseudo-inverse and

$$B = \begin{bmatrix} b_1(u_1) & \cdots & b_{n_u}(u_1) \\ b_1(u_2) & \cdots & b_{n_u}(u_2) \\ \vdots & \vdots & \vdots \\ b_1(u_{n_s}) & \cdots & b_{n_u}(u_{n_s}) \end{bmatrix}$$

$$\tilde{x}_v = \begin{bmatrix} \left(\sum_{i=1}^{k} V_{i1}\right)^{-1} \left(\sum_{i=1}^{k} V_{i1}\,{}^wt_{c_i}\right) \\ \left(\sum_{i=1}^{k} V_{i2}\right)^{-1} \left(\sum_{i=1}^{k} V_{i2}\,{}^wt_{c_i}\right) \\ \vdots \\ \left(\sum_{i=1}^{k} V_{in_s}\right)^{-1} \left(\sum_{i=1}^{k} V_{in_s}\,{}^wt_{c_i}\right) \end{bmatrix}$$

being V_{ij} the matrix computed according to (1) for the j-th sample provided by the i-th image.

C. Evaluation of Estimation Error by Reprojection

To evaluate the estimation error, the 3D DLO B-spline obtained in Section IV-B is reprojected on each image and the difference with respect to the input 2D spline provided by Section III-A is computed. Considering a generic 3D spline sample $q(u_j) = B\,q_v$, its homogeneous representation is provided by $\bar{q}(u_j) = [q(u_j)^T\ 1]^T$. The projected coordinates $\tilde{p}_{ij} = [\tilde{p}_{x_{ij}}\ \tilde{p}_{y_{ij}}]^T$ of the j-th spline sample on the i-th image plane can be written as

$$\tilde{p}'_{ij} = \begin{bmatrix} \tilde{p}'_{x_{ij}} \\ \tilde{p}'_{y_{ij}} \\ \tilde{p}'_{z_{ij}} \end{bmatrix} = A\,[{}^{w}R_{c_i}^{T} \mid -{}^{w}R_{c_i}^{T}\,{}^{w}t_{c_i}]\,\bar{q}(u_j)$$

$$\tilde{p}_{ij} = \begin{bmatrix} \tilde{p}_{x_{ij}} \\ \tilde{p}_{y_{ij}} \end{bmatrix} = \begin{bmatrix} \tilde{p}'_{x_{ij}}/\tilde{p}'_{z_{ij}} \\ \tilde{p}'_{y_{ij}}/\tilde{p}'_{z_{ij}} \end{bmatrix}$$

where

$$A = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

is the camera matrix containing the camera intrinsic parameters, such as the focal length f and center point coordinates c_x and c_y. Then, the overall error is provided by collecting all together in a single vector the error related to every single image, i.e. $e = [\cdots\ e_{ij}\ \cdots]^T$, $j = 1, \ldots, n_s$, $i = 1, \ldots, k$, where $e_{ij} = \|p_{ij} - \tilde{p}_{ij}\|$ is the distance between the corresponding initial spline sample provided by Section III-B and the projection on the image plane of the estimated 3D spline sample. Finally, the mean error norm $\|e\|_{n_s k} = \sqrt{e^T e}/(n_s\,k)$ can be used to evaluate the quality of the estimation result.

D. Online Reconstruction Optimization

In a general stereo setup, the two sensors are fixed and, as a consequence, their baseline can not be modified. In our setup, instead, the mobility of the robot can be exploited in order to find the best baseline and distance from the object corresponding to the minimum depth error. Indeed, both the baseline b and the distance from the target object z are responsible for the overall depth estimation error arising in triangulation methods, with the well-known relationship [9]:

$$\epsilon = \frac{z^2}{b\,f}\,\epsilon_d \tag{2}$$

where ϵ denotes the depth error, f the focal length of the camera and ϵ_d the disparity error (assumed to be within one pixel in the following). Thus, given a set of points in the 3D space p : $\{p_i = (x_i\,y_i\,z_i)^T, i \in [1, n_s]\}$, the optimization problem aiming at minimizing the depth error can be implemented, having the following cost function:

$$\min_{\delta_z, b} \frac{1}{n} \sum_{n=0}^{n_s} \frac{(z_i + \delta_z)^2}{b\,f}$$

where δ_z denotes the camera distance increment from the object, a value that can be either positive or negative.

This multi-variable optimization problem is subjected to a set of bounds and constraints that limit the admissible search space. The bounds are thus defined as :

$$b \geq 0$$

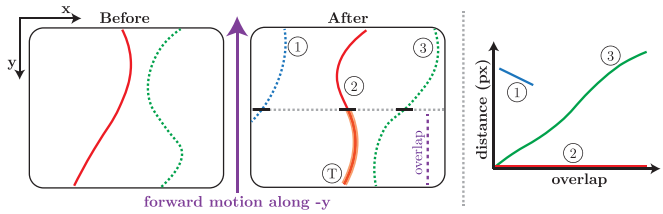

Fig. 5. Tracking the same DLO after a forward motion by exploiting a distance-based computation on the overlap area.

$$xs\delta_z^{\min} \leq \delta_z \leq \delta_z^{\max}$$

whereas the constraints are :

$$z_{\min} - z_i \leq \delta_z$$

$$z_i\,(-p_{x_i} + \sigma) \leq b\,f + \delta_z\,(p_{x_i} + \sigma) \tag{3}$$

$$b\,f + \delta_z\,(p_{x_i} + \sigma - w) \leq z_i\,(-p_{x_i} - \sigma + w) \tag{4}$$

where p_{x_i} is the row pixel value corresponding to the 3D coordinate p_i, σ is a safe offset in pixel coordinates to avoid regions of the image near the borders, w is the image width and $z_{\min}$ denotes the minimum distance of the camera from the 3D point. The solution to this minimization problem provides the optimal pair of baseline b and camera distance increment δ_z. Notice that (3) and (4) restrict the value of the parameters b and δ_z such that all the points p are inside the k images taken using the optimal parameters.

The optimization routine requires as input an initial guess of the depth values z_i. Thus, a coarse guess should be utilized or an initial execution of *DLO3DS* with fixed default parameters for b and δ_z is required for computing the initial guess. Moreover, in the case of tracking of the DLO shape (see Section IV-E), the values of the previous section can be used as an initial guess.

E. Tracking

In order to achieve a precise estimation of the DLO shape, *DLO3DS* is executed with camera samples captured in the proximity of sections of the DLO, e.g. the depth error is proportional to z (2). Thus, if the estimation of a long DLO shape is sought, a different approach is required. In this section, we described the steps employed for applying *DLO3DS* in a tracking framework, thus reconstructing the full 3D shape of a DLO combining individual estimations of small sections. In particular, after the estimation of a given section of the DLO, the camera is moved forward along the DLO principal direction and centered with respect to the estimated points. Thus, based on the overlap parameter n_o, a given percentage of previous points are still visible in the next DLO section and they are used for keeping track of the DLO under reconstruction, even in presence of multiple DLOs in the scene, as shown in Fig. 5.

At the end of the tracking performed along a DLO, the 3D points estimated for each segment of the DLO under analysis are collected in a unique vector in order to then obtain a single spline curve able to represent the overall DLO 3D shape.

Moreover, in order to further improve the estimation, these points are filtered to eliminate outliers and overlaps produced by subsequent acquisitions. To this end, the Locally Weighted Scatterplot Smoothing (Lowess) algorithm [31], a locally weighted

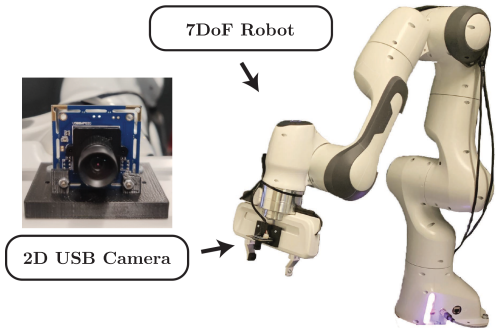

Fig. 6. Experimental setup composed of a Panda robot from Franka Emika and a low-cost eye-in-hand 2D USB camera.

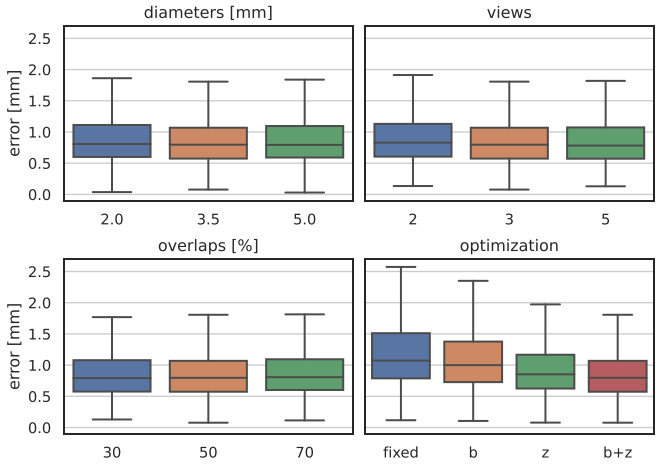

Fig. 7. Error distribution on the synthetic *test set* when varying a single parameter of *DLO3DS*. For the optimization plot: $fixed$ means fixed setup, b means just baseline, z means just distance, $b + z$ means both baseline and distance.

regression method that works by defining a window in the sample data, is applied for the final filtering of the points.

V. Experimental Validation

DLO3DS is validated experimentally employing a 7DoF robotic arm, the Panda from Franka Emika, equipped with an eye-in-hand 2D low-cost camera having a resolution of 640×480 pixels. The camera is both intrinsically and extrinsically calibrated, as shown in Fig. 6. The experiments are performed both with simulated and real data, in Section V-A and Section V-B respectively. Moreover, in Section V-C, *DLO3DS* is characterized in terms of processing time.

A. Evaluation in Simulation

To perform a proper evaluation of *DLO3DS*, ground truth data is needed. Considering that it is quite difficult to obtain an error-free 3D ground truth shape of a real DLO, synthetically generated data [32] is exploited to assert the *DLO3DS* performances. Thus, a *test set* of 10 randomly shaped reference synthetic DLOs of 0.8 meters in length is generated resembling the shape and appearance of real DLOs. They are accompanied by ground truth data in the form of 3D points describing their center line.

1) Influence of DLO3DS Parameters and DLO Diameter: The *test set* is rendered using three different reference diameters $\phi = 2.0$, 3.5, and 5.0 mm to analyze how the DLO thickness may affect the performance of *DLO3DS*. In addition, we analyzed the influence of the number of views, the percentage of overlap during the tracking (Section IV-E), and the contribution of the online optimization approach compared to a fixed stereo parameter setup or a partial optimization. When otherwise not specified, the default values are cable diameter 3.5 mm; number of views 3; overlap 50%; optimization of baseline and distance from the object ($b + z$). With default settings, the estimation mean error is 0.821 mm whereas the reprojection mean error is 0.731 pixels.

The box plots resulting from this analysis are depicted in Fig. 7. From the plots, it is possible to conclude that the diameter of the DLO does not play a major role in the estimation error. The same can be said for the overlap percentage, with the only remark that in a real estimation a bigger overlap may help to compensate for calibration errors. On the contrary, the slight drop in the error between 2 and 3 views is noticeable. Indeed, we commonly deploy *DLO3DS* using 3 views since the increase in the algorithm processing time is negligible and can be mostly compensated by its execution in masked time, as detailed in Section V-C.

Ultimately, online optimization does play a major role in bringing the interquartile range of the reconstruction error between 1 and 0.5 mm. The contribution of optimizing just the baseline corresponds to an error drop of 9 % compared to the fixed setup. Instead, the optimization of the camera distance provides a drop of 22 %. The joint optimization makes the error drop of 29 %. We claim that the major relative improvement of z as opposed to b compared to the fixed setup is due to the changing of the *virtual* baseline, i.e. the baseline virtually increases when the camera is moved closer to the object. Thus, in the z experiment there is an actual minor coupling with b making its result closer to the $b + z$ configuration.

2) Comparison With Baseline Methods: A comparison between *DLO3DS*, established methods like Semi-Global Matching (*SGM*) [7], and more recent approaches like *SISTER* [17] is provided by rendering the sample number 1 of the *test set* with different backgrounds and colors. For the estimation performed by *DLO3DS*, we used 3 views. Concerning the *SGM* method, we used just 2 views and we compute the matching cost one time via Census Transform (denoted as *CENSUS/SGM*) and a second time via a learned similarity measure [13], [33] (denoted as *MCCNN/SGM*). Finally, for *SISTER* we used 5 views as detailed in [17]. Aiming at a fair evaluation, in all the experiments the baseline was set to 25 mm, and the not-optimized fixed setup was employed, see Section V-A1.

Fig. 8 shows the computed depth images normalized between the min and max values of the ground truth one. Both *SISTER* and *SGM* provide as output a disparity image, thus we converted it into a depth image given the known baseline and focal length. Instead, *DLO3DS* provides as output just 3D points describing the DLO center line. In order to compute the depth image, the estimated 3D points and the colored mask of Section III-A are used to first estimate the radius of the DLO in world coordinates. Then, the original center line description is over-sampled and used to reconstruct the DLO surface keeping into consideration its radius. The result is a dense depth image of the DLO. For a fair comparison, the methods are evaluated only for what concerns

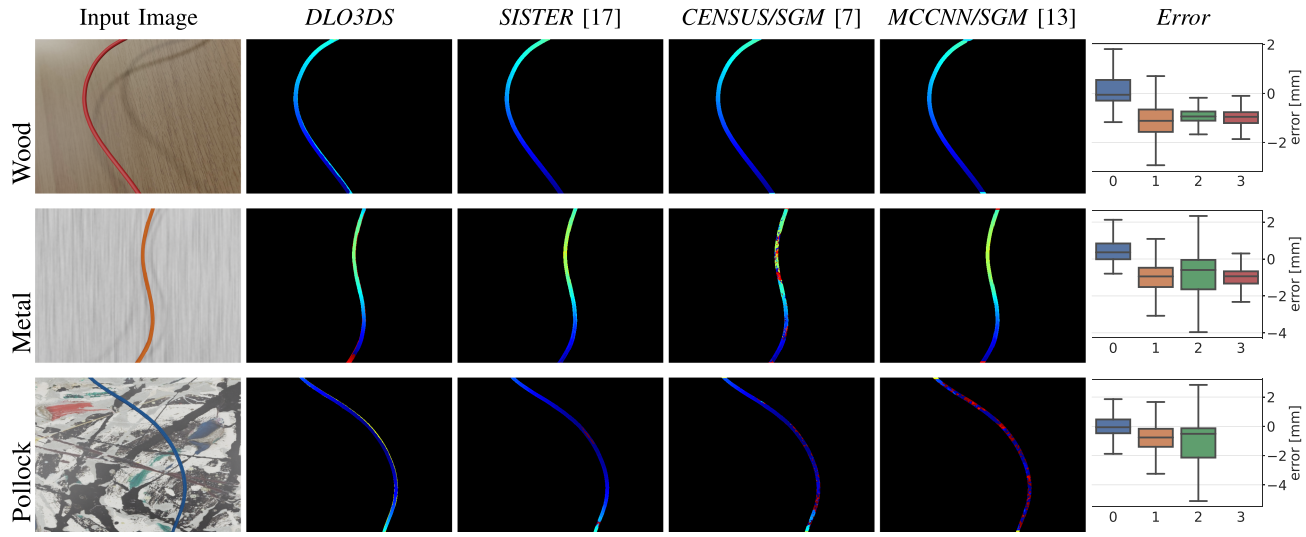

Fig. 8. Comparison with baseline methods in the form of depth images and boxplot. Plot legend: 0) *DLO3DS*, 1) *SISTER*, 2) *CENSUS/SGM* and 3) *MCCNN/SGM*. The display of the *MCCNN/SGM* boxplot in the third row is avoided due to large errors.

the depth values belonging to the DLO, the ground truth mask was used to select those points. The error between each method and the ground truth depth is computed by subtracting the latter from the first and it is shown using a box plot capturing the error distribution. From the figure, it is clear that *DLO3DS* provides an overall better estimate of the depth with wrong estimates only along the DLO boundaries due to prediction error in the segmentation mask.

B. Real-World Evaluation

To establish a boundary value of the estimation error in a real application, an experiment is performed using two types of purposely designed gripper fingers that, once closed, provide a hollow circle with a diameter of 6 and 10 mm respectively. First, *DLO3DS* is applied to estimate the DLO shape, then the center of the circumference is used as the reference frame for the generation of the motion: the robot should successfully follow the DLO without touching it, despite the shape of the DLO and changes in the z values. For the sake of generality, this experiment is performed both with electrical cables having a diameter of 3.5 mm and also with a different type of DLO, a polymeric hose for medical applications with an external diameter of 1.2 mm. The material of this hose is semi-transparent, such that it is almost invisible to commercial 3D sensors (including high-end ones) and laser scanners [5]. In Fig. 9, key-frames from a video sequence showing the experiment are reported. The cable of 3.5 mm is tested with the 10 mm gripper, while the hose with the one of 6 mm. Despite the complexity of the task, the cheap 2D camera used in this work is able to provide a reliable reconstruction of the sample objects, allowing for correct tracking without touching in both experiments.

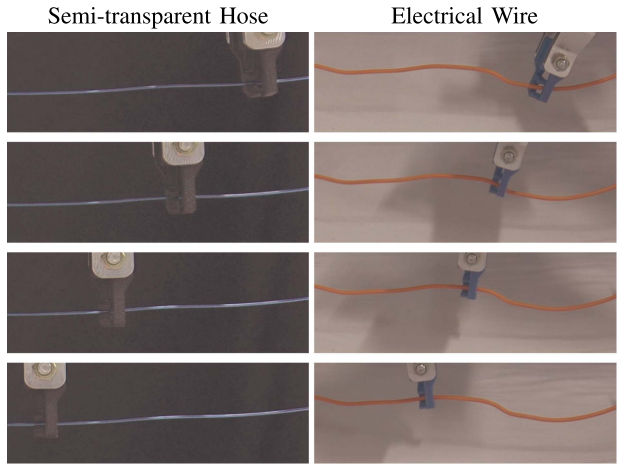

Fig. 9. Key-frames from a video sequence (available as supplementary material) showing the tracking test performed on DLOs of different types and diameters. Tester gripper diameter: black 6 mm, blue 10 mm.

C. Timings

The execution timings of *DLO3DS* are affected, other than the specific computing resources used, by the instance segmentation, modeling and selection performed in Section III, and by the triangulation procedure of Section IV. The timings of the first are mostly correlated to the choice of the image processing algorithm. By employing *FASTDLO* [4], 20 FPS are guaranteed for processing a single image when deployed on a workstation equipped with an Intel i9-9900 K CPU and Nvidia 2080Ti [4]. The performances' triangulation procedure of Section IV is affected by the number of points (n_s) at which the spline is evaluated. The following values are obtained for some configurations: $n_s = 10$, 7.5 ± 3.3 ms; $n_s = 20$, 19.5 ± 9.3 ms; $n_s = 40$, 27.2 ± 12.1 ms. Overall, *DLO3DS* provides competitive performances. It is worth mentioning that the data processing on a real setup can be mostly executed in masked time while the robot is moving toward the next pose.

VI. Conclusion

DLO3DS utilizes multiple 2D acquisitions for the accurate 3D shape estimation of DLOs. It is a fundamental tool for enabling the manipulation of DLOs by means of a robot without the need for expensive, bulky, and constrained 3D sensors. Thus, *DLO3DS* can be particularly useful in industrial applications

aiming for low-cost and effective solutions to complex manufacturing tasks involving the manipulation of cables, hoses, wires, ropes, and other similar objects.

DLO3DS in its current form deals with static scenes, i.e. the DLOs are still between the images acquisitions, and can be susceptible to the quality of the extracted splines. Thus, future activities will be devoted to addressing dynamic scenes and increasing robustness in cluttered conditions.

REFERENCES

[1] X. Jiang, K.-M. Koo, K. Kikuchi, A. Konno, and M. Uchiyama, "Robotized assembly of a wire harness in a car production line," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 490–495.

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey," *Int. J. Robot. Res.*, vol. 37, pp. 688–716, 2018.

[3] A. Caporali, R. Zanella, D. D. Greogrio, and G. Palli, "Ariadne+ : Deep learning–based augmented framework for the instance segmentation of wires," *IEEE Trans. Ind. Informat.*, vol. 18, no. 12, pp. 8607–8617, Dec. 2022.

[4] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "FASTDLO: Fast deformable linear objects instance segmentation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 9075–9082, Oct. 2022.

[5] K. P. Cop, A. Peters, B. L. Žagar, D. Hettegger, and A. C. Knoll, "New metrics for industrial depth sensors evaluation for precise robotic applications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5350–5356.

[6] A. Caporali, K. Galassi, and G. Palli, "3D DLO shape detection and grasp planning from multiple 2D views," in *Proc. IEEE/ASME Int. Conf. Intell. Mechatronics*, 2021, pp. 424–429.

[7] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[8] Y. Furukawa et al., "Multi-view stereo: A tutorial," *Found. Trends Comput. Graph. Vis.*, vol. 9, pp. 1–148, 2015.

[9] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[10] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," in *Proc. Int. J. Comput. Vis.*2002, pp. 131–140.

[11] D. Scharstein, T. Taniai, and S. N. Sinha, "Semi-Global Stereo Matching With Surface Orientation Priors," in *Proc. Int. Conf. 3D Vis.*, 2017, pp. 215–224.

[12] D. Hernandez-Juarez, A. Chacón, A. Espinosa, D. Vázquez, J. C. Moure, and A. M. López, "Embedded real-time stereo estimation via semi-global matching on the GPU," *Procedia Comput. Sci.*, vol. 80, pp. 143–153, 2016.

[13] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *J. Mach. Learn. Res.*, vol. 17, pp. 2287–2318, 2016.

[14] A. Seki and M. Pollefeys, "SGM-Nets: Semi-global matching with neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6640–6649.

[15] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5410–5418.

[16] D. Honegger, T. Sattler, and M. Pollefeys, "Embedded real-time multi-baseline stereo," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5245–5250.

[17] D. De Gregorio, M. Poggi, P. Z. Ramirez, G. Palli, S. Mattoccia, and L. Di Stefano, "Beyond the baseline: 3D reconstruction of tiny objects with single camera stereo robot," *IEEE Access*, vol. 9, pp. 119755–119765, 2021.

[18] A. Zeng et al., "Multi-view self-supervised deep learning for 6D pose estimation in the amazon picking challenge," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1386–1383.

[19] C. Hernandez et al., "Team Delft's robot winner of the amazon picking challenge 2016," in *Proc. RoboCup 2016: Robot World Cup XX 20*, 2017, pp. 613–624.

[20] S. Sharifzadeh, I. Biro, N. Lohse, and P. Kinnell, "Abnormality detection strategies for surface inspection using robot mounted laser scanners," *Mechatronics*, vol. 51, pp. 59–74, 2018.

[21] R. Zanella, A. Caporali, K. Tadaka, D. De Gregorio, and G. Palli, "Auto-generated wires dataset for semantic segmentation with domain-independence," in *Proc. IEEE Int. Conf. Comput., Control Robot.*, 2021, pp. 292–298.

[22] A. Caporali, M. Pantano, L. Janisch, D. Regulin, G. Palli, and D. Lee, "A weakly supervised semi-automatic image labeling approach for deformable linear objects," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 1013–1020, Feb. 2023.

[23] D. B. Camarillo, K. E. Loewke, C. R. Carlson, and J. K. Salisbury, "Vision based 3-D shape sensing of flexible manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 2940–2947.

[24] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3426–3433, Oct. 2018.

[25] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2372–2379, Apr. 2020.

[26] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proc. Med. Image Comput. Comput. Assist. Interv.*1998, pp. 130–137.

[27] J. Staal, M. D. Abrámoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Tran. Med. Imag.*, vol. 23, no. 4, pp. 501–509, Apr. 2004.

[28] V. Pătrăucean, P. Gurdjos, and R. G. Von Gioi, "A parameterless line segment and elliptical arc detector with enhanced ellipse fitting," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 572–585.

[29] D. De Gregorio, G. Palli, and L. Di Stefano, "Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Proc. 14th Asian Conf. Comput. Vis.*, 2018, pp. 662–677.

[30] A. Caporali, K. Galassi, G. Laudante, G. Palli, and S. Pirozzi, "Combining vision and tactile data for cable grasping," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, 2021, pp. 436–441.

[31] W. S. Cleveland, "Lowess: A program for smoothing scatterplots by robust locally weighted regression," *Amer. Statistician*, vol. 35, p. 54, 1981.

[32] M. Denninger et al., "Blenderproc," 2019, *arXiv:1911.01911*.

[33] D. Scharstein et al., "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. 36th German Conf. Pattern Recognit.*, 2014, pp. 31–42.

Open Access funding provided by 'Alma Mater Studiorum - Universit? di Bologna' within the CRUI CARE Agreement