Mechatronics 96 (2023) 103085

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

An approach based on machine vision for the identification and shape estimation of deformable linear objects[☆]

Pablo Malvido Fresnillo [a,*], Saigopal Vasudevan [a], Wael M. Mohammed [a], Jose L. Martinez Lastra [a], Jose A. Perez Garcia [b]

[a] FAST-Lab, Faculty of Engineering and Natural Sciences, Tampere University, Tampere, Finland
[b] Design and Fabrication in Industrial Engineering, University of Vigo, Vigo, Spain

ARTICLE INFO

Keywords:
Deformable linear object
Computer vision
DLO shape estimation
Self-critique system

ABSTRACT

The automation of processes that handle deformable materials, and in particular Deformable Linear Objects (DLOs), such as cables, ropes, and sutures; is a challenging task. Due to their properties, it is very difficult to predict the shape of these objects, making indispensable the use of perception systems for their manipulation. However, the detection of a DLO is a non-trivial task, and it can be even more complicated when additional considerations are made, such as detecting multiple DLOs, with small distances between them or even adjacent to each other, and with occlusions and entanglements between them. In this paper, a novel machine vision approach for estimating the shape of DLOs is proposed to address all these challenges. This approach processes the different DLOs in the image sequentially, repeating the following procedure for each of them. First, the DLO is segmented by examining the colors and edges in the image. Next, the remaining pixels are analyzed using evaluation windows to identify a series of points along the DLO's skeleton. These points are then employed to model the DLO's shape using a polynomial function. Finally, the output is evaluated by an unsupervised self-critique module, which validates the results, or fine-tunes the system's parameters and repeats the process. The performance of the system was tested with several wiring harnesses, detecting all their cables in homogeneous and complex backgrounds, with adjacent cables, and with occlusions. The results show an outstanding performance, with a successful shape estimation rate of more than 90% for some of the system configurations.

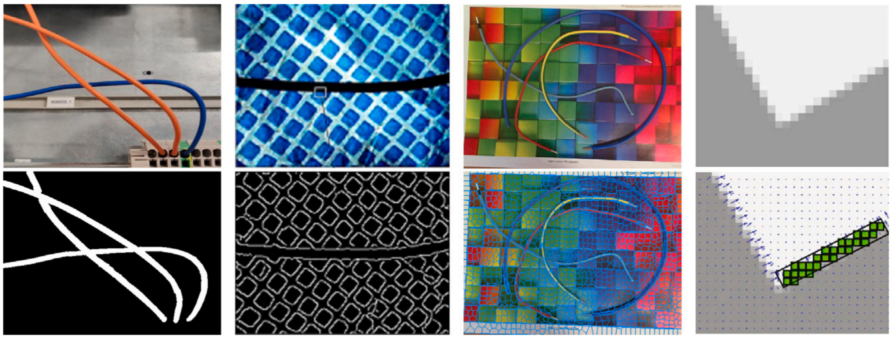
1. Introduction

Deformable Linear Objects (DLOs) such as cables, hoses, ropes, wiring harnesses or sutures, are very present in our daily life and are handled to perform many different tasks in industrial, medical, or household applications, among others. The manipulation of these objects seems simple and intuitive for humans thanks to our perception, dexterity, experience and knowledge about the environment. However, automating this task is very difficult and in most cases it is still performed manually [1]. This extra complexity, with respect to the manipulation of rigid objects, is due to the new challenges that deformable materials introduce to robotic manipulation. This includes the complication of sensing deformation, the high number of degrees of freedom of the DLOs, the complexity of non-linearity in modeling deformation [2], the dependency of the applied contact forces on the object deformation [3], etc.

Yin et al. [4] broke down robotic manipulation of deformable materials into three simpler problems: modeling, sensing and manipulation. A similar division was done by Zhu et al. [2], in which, besides the previous three, two additional problems were considered: planning and control. The approach presented in the current paper focuses on the sensing problem, aiming at developing a robust solution for the shape estimation of DLOs. Different sensing techniques can be used with this purpose, such as tactile sensing [5], force sensing [6], or proximity sensing [7], however machine vision is the most used technique as it provides global information about shapes on a large scale and does not require contact with the object. Within computer vision there is also a wide variety of approaches and possibilities, e.g., using 2D or 3D cameras, capturing colors or in grayscale, and a plethora of methods for analyzing the images, which gives flexibility for adapting to the specific requirements of the aim application.

[☆] This paper was recommended for publication by Associate Editor Garrett M. Clayton.
[*] Corresponding author.
E-mail address: pablo.malvidofresnillo@tuni.fi (P. Malvido Fresnillo).

https://doi.org/10.1016/j.mechatronics.2023.103085
Received 6 June 2022; Received in revised form 12 May 2023; Accepted 6 October 2023
Available online 16 October 2023



(a) Background subtraction [13]

(b) Edges detection [14]

(c) Superpixel segmentation [9]

(d) Other techniques [15]

Fig. 1. Machine vision techniques used as initial step for the shape estimation of DLOs.

The objective of this paper is to develop a machine vision system able to identify and estimate the shape of multiple DLOs in homogeneous (i.e., single color and texture, with no edges and no additional objects) and complex backgrounds. The system must be able to detect thin DLOs with small distances between them and must be robust against the occlusions caused by entanglements between the DLOs. Additionally, the parameters of the system must be adjusted automatically to maximize its performance with different light conditions and background colors.

The rest of the document is structured as follows: Section 2 includes a review of the existing computer vision techniques for the shape estimation of DLOs, identifying research gaps. Section 3 describes the developed system and its different modules. Section 4 presents the experimental evaluation of the developed system with different DLOs specimens and discusses about the obtained results. Finally, Section 5 reports the conclusions and some of the possible future lines of research.

2. State of the art

Many authors have tried to solve the problem of estimating the shape of DLOs using machine vision. Many different techniques, strategies and types of images have been used, leading to very different approaches that focus on different aspects depending on the requirements of the aim application. For instance, some of these approaches have to work in cluttered environments, so they are more tailored for being robust against occlusions of the DLOs [8] and complex backgrounds [9], but others are developed for applications that deal with small specimens or with small distances between them, requiring good performance against very thin [10] or adjacent [11] DLOs respectively. Another example are the tracking approaches [12], which need to be really fast in order to operate in real time and, normally, they use the configuration of the DLO in the previous time step as an input for the DLO shape estimation process. To present an overview of the state of the art, the reviewed approaches have been classified according to the initial step of their algorithms, differentiating them into four categories: background subtraction (or DLO segmentation), edges detection, superpixel segmentation, and other techniques (see Fig. 1).

Background subtraction: This technique, involves separating the foreground objects from the background in an image. Some authors refer to this operation as DLO segmentation, as its objective is to remove the background or noise from the image and isolate the objects of interest. One of the simplest background subtraction techniques is image differencing, which involves subtracting a reference or background image from the current image, which contains the evaluated DLOs, to obtain an image containing only the differences between the two. This technique is used in [13], taking an image with and without the DLO with a stationary camera. Then, a contour following algorithm is applied to the resultant binary image to identify the borders of the DLO. This algorithm examines the pixels of the DLO in a particular direction that alternates between horizontal and vertical upon detection of a boundary.

Another simple DLO segmentation method is color filtering, where all colors except for those that correspond to the objects of interest are filtered out. This technique is applied in [8] to segment a rope placed over a green background. The resultant point cloud is then discretized in nodes and, finally, the shape of the DLO is estimated using Coherent Point Drift (CPD), which makes the approach robust against occlusions. A very similar solution was presented in [12], where the background is also subtracted with a color filter, and the DLO is modeled using Constrained Deformable Coherent Point Drift (CDCPD), which consists of an enhanced CPD with additional constraints to ensure that the DLO never passes through itself and is never inside an obstacle. Both these approaches work with point clouds captured with 3D cameras.

Another example of color filtering is presented in [14]. In this approach, the obtained binary image is processed by a Fully Convolutional Network (FCN) to obtain keypoints of the DLO and, finally, the position of these points is corrected using a geometric fine-tuning. The geometric analysis of the segmented DLO is a common practice, which can also be found in other approaches. For instance, in [15], a geometric optimal control algorithm is utilized to estimate the shape of a segmented elastic rod by evaluating the position of a set of equidistant points along the rod.

These background subtraction methods are fast and simple to implement and work well in many situations. However, for more complex or dynamic environments the use of more advanced methods such as machine learning algorithms may be required. In particular, Convolutional Neural Networks (CNNs) are a widely used technique for object instance segmentation [16], and numerous approaches take advantage of them, such as [17–19], and [20].

The approach proposed in [17] employs a CNN to segment wiring harnesses, classifying the pixels of the image into eight different classes, including the wire bundles, the background, and other elements of the background platform. In [18], the authors presented a methodology for identifying entangled ropes using two CNNs. The first network was utilized for segmenting the rope from a complex background, while the second CNN (a Yolov3 model [21]) was used to determine its cross points in order to analyze which part of the DLO is on top when there are entanglements. In [19], an approach for identifying the pixels of power cables in 2D color images with complex backgrounds is presented. This is achieved using a CNN based model composed of an encoder–decoder block that extracts features from the image, a prediction module for object classification and key points regression, and a DBSCAN clustering module, which merges the results of the network and models each of the cables. Finally, the approach proposed in [20] exploits CNNs to introduce one of the fastest algorithms for identifying DLOs in the current state-of-the-art, with a processing rate higher than 30 FPS (Frames per Second). The employed methodology starts by subtracting the background using a CNN. Then, a set of vertices are sampled from the segmented image, characterizing their orientation with a second CNN. After this, the relation between the vertices is evaluated, creating a graph representation of the DLOs, which is finally used to analyze the intersection areas and identify each DLO instance.

Although CNNs are the most popular machine learning method for image segmentation, some authors have utilized other techniques for this purpose. This is the case of the approach proposed in [22], where a hierarchical Self-Organized Map (SOM) is used to classify and segment fur tails based on texture properties. Then, the skeleton of these furs is determined using a sequential thinning algorithm. The adoption of skeleton thinning after image segmentation is a usual strategy, which has been used in other approaches in the literature, including [23,24]. Nevertheless, in [24], this strategy is further enhanced by implementing additional operations that link the identified DLO segments, making the algorithm robust against occlusions.

In addition to the aforementioned approaches, various techniques have been proposed for background subtraction in 3D images. One of these techniques is Region of Interest (ROI) intensity filters, which is used in [25] to remove the background from the point cloud of a wiring harness. Then, the segmented point cloud, which is modeled using a Gaussian Mixture Model (GMM), is processed using an enhanced Structure Preserved Registration (SPR) method to track the branched DLO. Similar to [8,12], this approach shows excellent results tracking DLOs in the presence of occlusions. Another commonly employed method for subtracting the background of 3D images is plane fitting, which leverages depth information. This technique was utilized in the approach presented in [26], which after isolating the DLO point cloud, employs a region-growing method to over-segment it in a set of nodes. These nodes are then sorted by minimizing a cost function that relies on some geometrical features to reveal the structure of the DLO. An alternative way to process the segmented 3D point cloud of a DLO is presented in [27]. The approach involves sorting the points along the length of the DLO, performing a linear regression skeletonization with the sorted points, and finally, generating a kinematic multibody model of the DLO by determining the joints of its skeleton in an iterative procedure.

Edges detection: There is a wide variety within the approaches that start the DLO identification process by detecting the edges in the image. This diversity lies not only in the methods for extracting the edges but also in the approaches for analyzing them. One of the most popular algorithms for detecting the edges in an image containing DLOs is the Canny edge detector. This is used in [28] to extract the edges of an image containing an underwater cable with a complex background. After that, the edges of the cable are identified by classifying them with Multilayer Perceptron (MLP) and Support Vector Machine (SVM) classifiers. Finally, a Hough transform is used to locate the two border lines of the cable. Almost the same strategy is followed in [29], where Canny and Hough Transform are used to obtain first, the image edges and then, the cable lines. A Canny edges detector is also used in [10] to model extremely thin DLOs, such as sutures or tiny ribbons. This approach merges the edges detected in images captured by different cameras. Subsequently, the potential DLO configurations are represented with an energy function that incorporates the error of projecting all the computed edges in each image. The most likely DLO configuration is obtained by minimizing this energy function. A similar approach is used in [30], where in each of the two images taken by stereo vision, the edges are extracted and used to calculate the skeleton line of a wire. The resulting points from one image are projected onto the other, and the results are merged to determine the actual DLO points.

An enhanced version of the Canny edges operator is presented in [31], in which the spurious edges are removed by adding an Artificial Neural Network (ANN) that classifies the edges as spurious or not. Then, the shape of the DLO is estimated by template matching, using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method [32] to minimize the error between the detected edges and a deformable template. This template uses the Boundary Element Method (BEM) to model its deformation. Deformable template matching was also used in [33] to detect non rigid objects, but in this case using triangulated polygons as templates. This approach defines an energy function that assigns a cost to each possible transformation of the deformable template to match the detected shape. The template shape that best fits the analyzed object is obtained by minimizing this function.

The effectiveness of four different edge detection operators, namely Difference, Roberts, Sobel and Laplacian-of-Gaussian operators, for identifying the boundaries of a cable in a complex background was evaluated in [34]. Results showed that the Sobel operator performed the best. After this, the cable shape was analyzed using evaluation windows, propagating the cable lines forward. This method starts placing a window on a known point of the cable, and subsequent windows are generated adjacent to the previous one and concentric to the cable direction.

A distinct DLO edge detection approach was used in [35], which involves the application of clustering. This methodology identifies the regions with the higher gradient between clusters and designates them as edges. Two clustering algorithms were tested, namely minimal-spanning tree and K-means clustering, obtaining better results with the first one. After this, an eigenvector straight line fitting method was used to reduce the noise caused by the complex background and, finally, the actual cable borders were detected by evaluating the parallelism between edges. Eigenvectors are a powerful tool to determine the direction of the DLOs, and they have also been used in various other approaches. For instance, in [36,37] the eigenvalues of the Hessian of the image are utilized to detect vessels in medical images.

Superpixel segmentation: There is another group of approaches which, instead of starting with a background subtraction or edges detection operation, perform a superpixel segmentation of the image. This reduces the computation time of the following operations, as they can focus on the analysis of meaningful regions (i.e., superpixels) instead of the whole set of pixels. This is the case of [9], where a superpixel segmentation is performed to build Region Adjacency Graph (RAG), and then, an iterative algorithm tries to find the best path through the RAG for connecting the DLO endpoints. A similar approach is followed in [11], where after a multi-scale superpixel segmentation of the image, the most likely sequence of superpixels is found by minimizing a cost function using dynamic programming.

Other techniques: Although most of the vision-based DLO recognition approaches can be classified into one of the three previous categories, there are others that follow a different strategy and do not start the image analysis with any of those operations. Some examples of this are [38,39]. In [38], a methology for identifying line segments and circular and elliptical arcs in an image is presented. Additionally, this algorithm was used in [9] to recognize wires with good results. In this approach, the initial line and arc candidates are identified through a process of region growing and curve growing.

Table 1
Characteristics and capabilities of published vision-based DLO detection approaches (1/2). Ent.: Entanglements, Occ.: Occlusions, Adj.: Adjacent, NT: Not tested.

Ref	Test specimens	DLOs number	Size (D/L)	DLOs color	Background	Ent.	Occ.	Adj.
Background subtraction								
[13]	Wires	1	Small	Grayscale	Homogeneous	NT	NT	NT
[8]	Rope	1	Big	1	Homogeneous	✓	✓	NT
[12]	Rope	1	Medium	1	Homogeneous	✓	✓	NT
[14]	Rope	1	Small	1	Homogeneous	NT	NT	NT
[15]	Elastic rod	1	Small	1	Homogeneous	NT	NT	NT
[17]	Wire harness	Multiple	Medium	1	Homogeneous	NT	✓	NT
[18]	Ropes	1 and 2	Medium	1	Complex	✓	NT	NT
[19]	Power cables	Multiple	Tiny	1	Complex	NT	NT	NT
[20]	Wires	Multiple	Small	Multiple	Complex	✓	NT	✗
[22]	Furs	Multiple	Big	1	Complex	✗	✗	✗
[23]	Ropes	1	Medium	1	Homogeneous	✓	NT	NT
[24]	Wires	1	Small	1	Complex	✓	✓	✗
[25]	Wire harness	Multiple	Medium	1	Homogeneous	NT	✓	NT
[26]	Cables, ropes	1	Medium	1	Homogeneous	✓	NT	NT
[27]	Vacuum hose	1	Big	1	Complex	NT	NT	NT
Edges detection								
[28]	Underwater cable	1	Medium	1	Complex	NT	NT	NT
[29]	Pipe	1	Big	Grayscale	Homogeneous	NT	NT	NT
[10]	Suture, ribbon	1	Tiny	1	Homogeneous	NT	NT	NT
[30]	Wires	1	Small	Grayscale	Homogeneous	NT	NT	NT
[31]	Deformable obj	–	–	Grayscale	Complex	NT	✓	NT
[33]	Deformable obj	1	Big	Grayscale	Complex	NT	✓	✓
[34]	Cables	1	Small	Grayscale	Complex	NT	NT	NT
[35]	Underwater cable	1	Big	Grayscale	Complex	NT	✓	NT
[36]	Vessels, wires[a]	Multiple	Small	Multiple	Homogeneous	✓	NT	NT
[37]	Vessels, wires[a]	Multiple	Tiny	Multiple	Homogeneous	✓	NT	NT
Superpixel segmentation								
[9]	Wires	Multiple	Small	Multiple	Complex	✓	NT	✗
[11]	Curved objects	Multiple	Medium	Multiple	Complex	NT	✗	✓
Other techniques								
[38]	Curves, wires[a]	Multiple	Small	Multiple	Homogeneous	✓	NT	NT
[39]	Vessels	Multiple	–	Grayscale	–	NT	✓	NT

[a] *The approaches have been also tested with wires in [9].*

Then, a probabilistic method is applied to eliminate false detections and, finally, a model is selected for the remaining candidates. Regarding [39], an energy function minimization technique is used to determine the deformable contour of an object, based on the intensity gradient of the pixels and the contour curvature. To speed up the process, the contours are just evaluated inside successive evaluation windows, whose positions and dimensions are determined based on some reference points provided by the user.

The main characteristics and capabilities of the vision-based DLO detection approaches reviewed in this section are summarized in Tables 1 and 2. Analyzing these tables, it can be seen that most of these approaches focuses on tracking applications, prioritizing a fast computation, but there is lack of solutions that can detect multiple adjacent tiny DLOs, and that are robust against entanglements (or at least that have demonstrated good performance in these situations).

3. DLO shape estimation system

3.1. Structure of the system

In this paper, a system for the identification and shape estimation of DLOs from 2D RGB images is presented. This system is composed of five modules, and its structure and workflow can be seen in the activity diagram of Fig. 2. It receives as inputs the RGB image of the DLOs to model and some predefined knowledge about the DLOs, like their diameter and color; and returns the estimated shape of all the DLOs as polynomial functions. The code of the system is publicly available.[1]

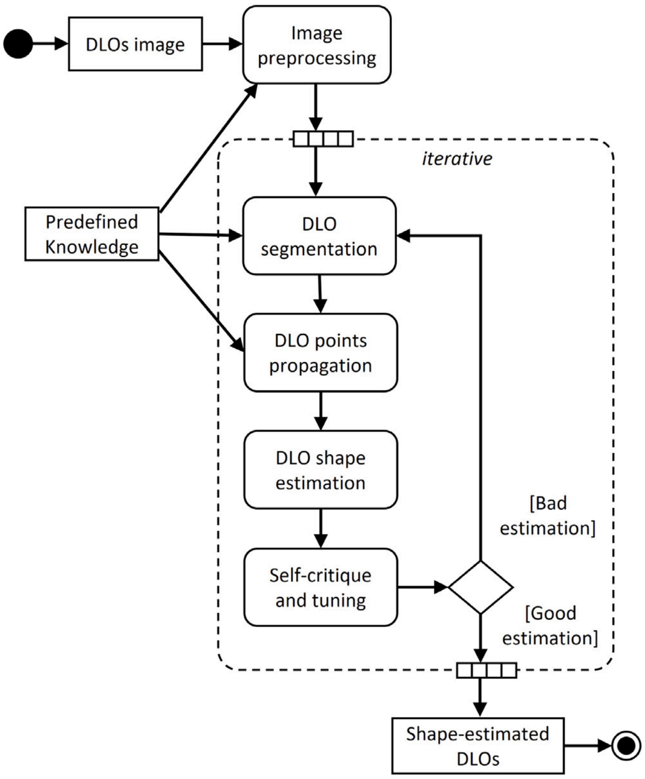


Fig. 2. DLO shape estimation system UML activity diagram.

[1] https://github.com/pablomalvido/DLO_shape_estimator_TAU

Table 2
Characteristics and capabilities of published vision-based DLO detection approaches (2/2).

Ref	Approach	Prev info	Outcome	Camera	Speed
	Background subtraction				
[13]	Image differencing + contour following algorithm	None	DLO base points	2D grayscale	< 10 ms
[8]	Color filtering + pointcloud discretization + CPD	Prev state	Skeleton points	3D stereo	< 100 ms
[12]	Color filtering + Enhanced GMM-EM regularization + enhanced CDCPD	Prev state	Skeleton points	3D RGBD	< 100 ms
[14]	Color filtering + FCN + geometric finetuning	Training	Skeleton points	RGB + lidar	–
[15]	Background subtraction + geometric optimal control	Endpoints	3D skeleton	2D RGB	< 1 s
[17]	CNN objects detection and segmentation	Training	Points of interest	2D RGB	< 100 ms
[18]	DLO segmentation (CNN) + cross points determination (CNN)	None	B-Splines	2D RGB	–
[19]	CNN pixel classification + DBSCAN clustering	Training	Linear segments	2D RGB	< 100 ms
[20]	DLO segmentation (CNN) + vertices sampling + edges sampling + graph analysis	Training	DLO points + overlapping info	2D RGB	< 100 ms
[22]	Object classification and segmentation (SOM) + sequential thinning skeleton function [40]	Training	Skeleton points	2D RGB	–
[23]	Background subtraction + thinning skeletonization + graph structure analysis + intersections analysis	DLO diameter	Topological model	3D stereo	–
[24]	Background subtraction + skeletonization + contour extraction + DLO fitting and pruning + merging	None	DLO points	2D RGB	< 1 s
[25]	3D background subtraction (ROI intensity filter) + enhanced SPR method	WH info	Linear segments	3D stereo	< 10 s
[26]	3D background subtraction (plane fitting) + DLO discretization by region-growing + cost function minimization to find the correct segments order	Training cost function weights	Linear segments	3D RGBD	–
[27]	3D background subtraction + Pointcloud sorting + skeletonization + Joints estimation	None	Multibody model	3D stereo	–
	Edges detection				
[28]	Edge detection (Canny) + classification (MLP and SVM) + lines detection (Hough transform)	None	Border lines	2D RGB	< 100 ms
[29]	Edge detection (Canny) + lines detection (Hough transform)	None	Border lines	2D RGB	< 100 ms
[10]	Edge detection (Canny) + local minimization of an energy function of the DLO	Training + init config	Vertices + edges + material frames	3D stereo	–
[30]	Edges detection + skeleton line calculation + stereo merging	None	3D skeleton	3D stereo	–
[31]	Edge detection (Canny + NN) + Template-based tracking using the BFGS method to minimize the error	BEM model	Object template (BEM)	2D grayscale	< 1 s
[33]	Deformable template matching + transformations energy function minimization	Template	Triangulated polygons	2D grayscale	> 1 min
[34]	Edge detection (Sobel) + DLO borders intersection with adaptative tracking windows	Thickness + prev state	Borders + linear regression	2D grayscale	–
[35]	Edges detection (clustering) + parallel contours evaluation	None	Axis line	2D grayscale	< 100 ms
[36]	Multi-scale filtering + Hessian eigenvalues analysis	None	Filter (visual)	DSA, MSA[a]	–
[37]	Ridges detection and grouping by Hessian eigenvalues analysis	None	Segmentation	2D RGB	–
	Superpixel segmentation				
[9]	Superpixel segmentation + biased random walks	Endpoints	B-Splines	2D RGB	< 1 s
[11]	Multiscale superpixel segm + cost function to group them	None	Linear segments	2D RGB	–
	Other techniques				
[38]	Candidates selection (region growing), validation and modeling	None	Lines and arcs	2D grayscale	–
[39]	Cost function minimization by dynamic programming in successive evaluation windows to determine the contour	Reference points	Contour segmentation	DSA, MRI[a]	–

[a] *The images taken by these cameras have been analyzed as 2D grayscale images.*

The first module performs a preprocessing of the image, preparing it for the following operations. In this preprocessing step, the resolution of the image is adjusted to have a certain number of pixels per DLO diameter (${}^{px}/_{D}$). In case there are DLOs of different diameters in the image, the smallest (D) will be considered for the calculation, ensuring always a minimum number of pixels per DLO diameter. For resizing the image it is necessary to know the pixels to millimeters conversion factor (${}^{mm}/_{px}$) in the plane of the DLOs. This information is obtained by checking the number of pixels of a reference structure situated in the same plane, for instance a connector in one of the cable ends. The dimensions and location of this reference structure is part of the predefined knowledge received by this module.

$$\begin{pmatrix} width \\ height \end{pmatrix}_{resized} = \begin{pmatrix} width \\ height \end{pmatrix} \cdot \frac{D}{{}^{mm}/_{px} \cdot {}^{px}/_{D}}$$

After this preprocessing, the image is ready to start recognizing the shape of the DLOs. This is managed by a group of modules that analyzes each DLO separately, thus, the process is repeated for all the DLOs of the image. First, A binarization of the image is performed, segmenting the DLOs of the required color (the color of the DLOs can be repeated). The segmentation algorithm is composed by two main operations, an enhanced color filter and a Canny edge detector, and finally their

Algorithm 1 DLO segmentation algorithm

```
procedure rgbDif(R, G, B, c)
    $c\_dif \leftarrow \sqrt{(R - c[0])^2 + (G - c[1])^2 + (B - c[2])^2}$
    return $c\_dif$
procedure colorFilter($img, c, c\_all, thr_1, thr_2$)
    $Bin \leftarrow$ Empty image of size $[img.width, img.height]$
    $r, g, b \leftarrow$ RGB channels of $img$
    for $x \leftarrow 0$ to $img.width$ do
        for $y \leftarrow 0$ to $img.height$ do
            $Bin[x, y] \leftarrow 0$
            $r_i, g_i, b_i \leftarrow r[x, y], g[x, y], b[x, y]$
            if $|r_i - c[0]| < thr_1$ and $|g_i - c[1]| < thr_1$ and $|b_i - c[2]| < thr_1$ then
                $cd \leftarrow$ call $rgbDif(r_i, g_i, b_i, c)$
                $c\_pixel \leftarrow$ True
                for each $c2 \in c\_all$ do
                    if $c2 \neq c$ then
                        $cd2 \leftarrow$ call $rgbDif(r_i, g_i, b_i, c2)$
                        if $cd2 \cdot thr_2 < cd$ then
                            $c\_pixel \leftarrow$ False
                            break
                if $c\_pixel$ then
                    $Bin[x, y] \leftarrow 1$
    return $Bin$
procedure Main($img, c, c\_all, thr_1, thr_2, thr_3, px/D$)
    $ColBin \leftarrow$ call $colorFilter(img, c, c\_all, thr_1, thr_2)$
    $EdgeBin \leftarrow$ CannyEdges($img$)
    $EdgeBinDil \leftarrow$ Dilate($EdgeBin, thr_3, px/D$)
    $Segm \leftarrow EdgeBinDil \cap ColBin$
    return Segm
```

results are merged. This algorithm is described in Algorithm 1, where c is the RGB color code of the DLO to segment, c_all is a list with the RGB color codes of all the DLOs, and thr_1, thr_2 and thr_3 are three parameters that determine how permissive the segmentation filter is. The RGB color codes are part of the predefined knowledge.

The color filter first splits the image into its color channels (red, green and blue). Then, for each pixel of the image, it checks if the difference between its color and the reference color c is, for all its color channels, lower than a threshold value (thr_1). Thus, if this threshold is very low, just a few of the DLO pixels will be recognized, but if it is too high, many similar colors will pass the filter, leading to a bad segmentation where it is difficult to recognize the DLO. To overcome this problem, a second filter is applied, which checks if the detected colors are more similar to any of the other known colors of the image (c_all). This allows the first filter to be more permissive, capturing more pixels of the DLO in question, but at the same time avoiding confusions with other similar colors. Moreover, a second threshold parameter (thr_2) is used to dynamically modify how restrictive the filter is with other colors.

On the other hand, the Canny edge detector is used to identify all the edges of the original image. Then, a dilation operator is applied to the resultant binary image, expanding the parallel border lines of each DLO until they get in contact, creating filled lines for all the DLOs of the image. To achieve this, the dilation kernel size is determined based on the number of pixels per diameter of the DLOs (px/D) and a reconfigurable value (thr_3) that can slightly increase or decrease this size to refine the DLO segmentation. Both operators, the color filter and the edge detector, can work as standalone segmentation systems for homogeneous backgrounds and not adjacent DLOs. However, when their results are merged and the intersection of both binary images is selected, a more robust segmentation is achieved, effectively reducing captured noise. This approach enables the detection of adjacent DLOs in complex backgrounds, even when the background and the DLO have the same color.

Then, the segmented binary image is sent to the next module, which analyzes it and determines the pixels of the skeleton line of the DLO. This can be done with two different algorithms, named Forward DLO propagation (FWP) and Backward DLO propagation (BWP), described in Sections 3.2 and 3.3 respectively. After this, the skeleton points of the DLO are sent to the DLO shape estimation module, in which its shape is modeled by polynomial regression. The order of the polynomial is determined by cross-validation, using 70% of the points for training and 30% for testing. The maximum possible order of the polynomial was set as eight.

Finally, the last module evaluates the results obtained based on several indicators. This module can either validate the result or reject it, providing a critique and fine-tuning the system parameters in the second case. In case of rejection, the DLO identification process is repeated again, with the adjusted parameters, from the DLO segmentation module. This module, the evaluated indicators and the corrective actions, are explained in detail in Section 3.4.

3.2. Forward DLO propagation (FWP)

The name of this algorithm comes from the principle followed to determine the skeleton points of the DLO, shown in Fig. 3. The algorithm starts from the initial point of the DLO, whose position must be known beforehand, and from there, it starts propagating the DLO points forward, following the DLO direction. The full process can be seen in Fig. 4.

The first step of this algorithm is determining the initial point of the DLO. The theoretical initial point is part of the predefined knowledge of the system, and it is defined as a distance from the reference structure mentioned in Section 3.1. In the case of wiring harnesses, the reference structure would be the main connector, and the initial points of the cables would be calculated with the distance between the corner of the connector and the cables insertion points. However, in order to correct misalignments, the initial point considered by the algorithm is the closest segmented pixel to the theoretical initial point.

Then, an iterative process starts, propagating the points until the end of the image or the end of the DLO. To calculate each new point of the DLO skeleton, a rectangular window is created from the last calculated point (the initial point in the first iteration), and the pixels inside it are evaluated. The window size depends on the px/D of the DLO, and it can be modified dynamically. All the segmented pixels located in the window borders are stored as potential new points, and the pixel in the intersection of the window border with the line that joins the two previous points is stored as the direction point. Then, the outcomes of the window evaluation are analyzed. If any potential new point was found in the window border, the selected one will be the closest to the direction point (Fig. 3(b)). In case, no points are found in the border, the segmented pixels inside the window are checked, and if any are detected, the one selected is again the closest one to the direction point. Finally, if no segmented pixels are detected neither on the border nor inside, the previously calculated direction point is considered the new DLO point (Fig. 3(c)). This makes the algorithm robust against small occlusions. This process continues, propagating the DLO points forward in each iteration, until the end of the image is reached, or until more than a certain number (mp_{max}) of consecutive iterations have not detected any point in their evaluation windows, which would mean that the DLO has ended.

3.3. Backward DLO propagation (BWP)

Unlike FWP, the BWP algorithm does not require to know the initial DLO points to start determining its points. Instead, it starts analyzing the image from the side where the DLOs end, and their points are propagated backward until reaching the initial points, that are just used to identify the recognized DLOs. The full process is described in Algorithm 2, and the working principle is shown graphically in Fig. 5.

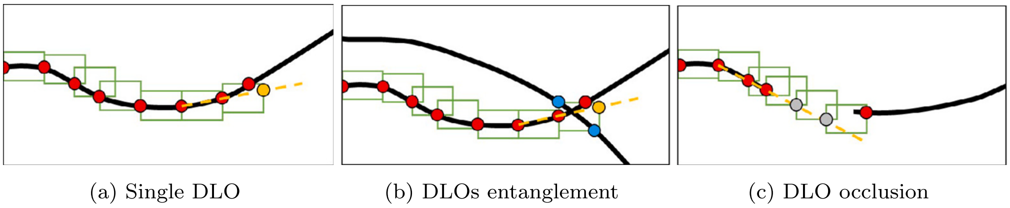

(a) Single DLO (b) DLOs entanglement (c) DLO occlusion

Fig. 3. FWP working principle in different scenarios. Green rectangles: pixel evaluation windows, yellow dashed lines: calculated points direction, red circles: skeleton points of the DLO, yellow circles: DLO direction points, blue circles: non-selected potential skeleton points, gray circles: skeleton points selected during DLO occlusions.

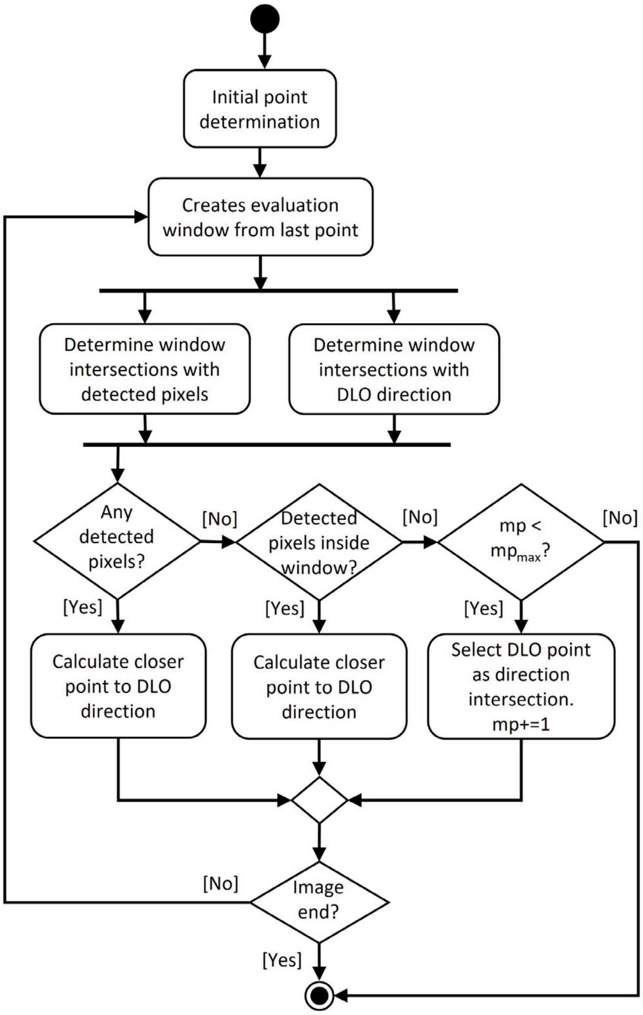

Initial point determination
Creates evaluation window from last point
Determine window intersections with detected pixels
Determine window intersections with DLO direction
Any detected pixels?
[No]
Detected pixels inside window?
[No]
mp < mp_{max}?
[No]
[Yes]
[Yes]
[Yes]
Calculate closer point to DLO direction
Calculate closer point to DLO direction
Select DLO point as direction intersection. mp+=1
[No]
Image end?
[Yes]

Fig. 4. FWP UML activity diagram.

The algorithm is structured in four main steps, and it receives as arguments the segmented binary image (img), the amount of segmented pixels in the binary image (bin_p), the number of DLOs of the analyzed color (n), its theoretical initial point ($init$), the size of the evaluation window (ws), and the maximum number of consecutive windows with no detected points (consecutive missed pixels, mpc_{max}). In the first step, all the potential DLOs are calculated using the FWP algorithm. However, the propagation, in this case, happens backward and it starts from every segmented pixel that has not been captured by the evaluation windows of any other potential line ($[x, y] \notin CP$). Some of these potential lines are composed of isolated pixels that passed the segmentation filters by mistake. Therefore, to exclude these points from the analysis, just the lines that capture more than 5% of the segmented pixels are considered as potential lines (PL).

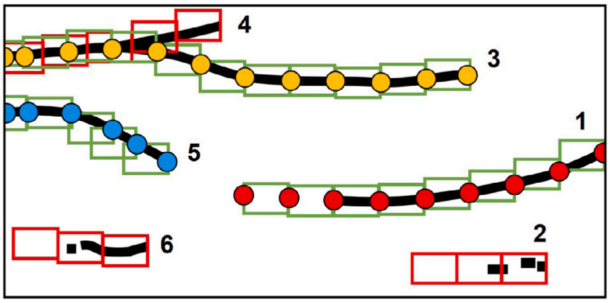


Fig. 5. BWP working principle. The figure shows six potential DLOs. The rectangles represent the evaluation windows, they are green if the line is accepted and red if it is discarded. The circles represent the potential DLO points and they have a different color for every segment. Lines 2 and 3 are discarded because they capture less than 5% of the image detected pixels. Line 4 is discarded because more than 80% of their pixels are also captured by line 3. Line 1 and 5 are joined into a single line. Finally, the recognized DLOs are 1–5 and 3.

Then, in the second step, the repeated potential lines are removed. It is considered that a line is repeated when it shares at least 80% of the points captured by their evaluation windows with any other potential line. In this case, the line with less points is eliminated, obtaining the list of independent lines (IL).

Due, to occlusions and entanglements with other DLOs, segments of the same DLO can be detected as different individual lines. The next step of the algorithm addresses this issue by joining the independent lines that belong to the same DLO. It is considered that two lines are part of the same DLO when the x coordinate of the last pixel of the first line is smaller than the x coordinate of the first pixel of the second line, the distance between them is smaller than 50 mm, and the angles difference between both lines is smaller than 35 degrees. If for a line there is more than one candidate for joining, the closest one is selected. The lines are then merged in a recursive way, so if a DLO is composed by more than two independent lines, the successive lines will be added to the already merged line until it is completed. At the end of this step, a list with all the merged lines is obtained (ML).

Finally, the fourth step selects the n final DLOs (FL) as the merged lines that have captured a higher number of segmented pixels. If there is more than one DLO of the analyzed color ($n > 1$), the retrieved line (L) will be the FL that is closer to the theoretical initial point. This distance is computed at the x coordinate of the initial point. Therefore, the shape of each FL is estimated (regression lines RL), and the y coordinate at this x is calculated with the resultant polynomial function. Unlike, the FWP algorithm, here the initial points are just used to identify the detected DLOs, being able to work even when the initial points are occluded.

3.4. Self-critique tuning module

There are many factors that can affect the DLO shape estimation process, such as the lightning, the colors and sizes of the DLOs, the distance between them, the background colors, entanglements between DLOs, or partial occlusions of the DLOs. The influence of some of these

Algorithm 2 BWP algorithm (Part 1)

```
procedure GetPotentialLines($img, bin\_p, n, ws, m$)
    $img2 \leftarrow$ Mirror $img$
    for $x \leftarrow 0$ to $img2.width$ do
        for $y \leftarrow 0$ to $img2.height$ do
            if $img2[x, y] = 1$ and $[x, y] \notin CP$ then
                $bp, cp, success \leftarrow$ call $FWP(img2, [x, y], ws, m)$
                $CP \leftarrow$ append $cp$
                if $len(cp)/bin\_p > 0.05/n$ and $success$ then
                    $PL \leftarrow$ append $\{bp, cp\}$
    return $PL$
procedure RemoveRepeatedLines($PL$)
    $PL \leftarrow$ sort by $len(cp)$
    $IL \leftarrow PL[0]$
    for each $pl \in PL[1:]$ do
        for each $il \in IL$ do
            if $len(pl.cp \cap il.cp)/len(pl.cp) \geq 0.8$ then
                $rep \leftarrow$ True
                break
        if not $rep$ then
            $IL \leftarrow$ append $pl$
    return $IL$
procedure JoinLines($IL, {}^{mm}/_{px}$)
    $i, j \leftarrow 0$
    for each $il \in IL$ do
        for each $il2 \in IL$ do
            $dp \leftarrow 50/{}^{mm}/_{px}$
            if $il[-1].x > il2[0].x$ and $|il[-1], il2[0]| \leq dp$ then
                if $angle\_dif(il, il2) < 35deg$ then
                    $JL'[i] \leftarrow$ append $\{l2 : j, D : |il[-1], il2[0]|\}$
            $j \leftarrow j + 1$
        if $i \in JL'$ then
            $JL[i] \leftarrow JL'[i].l2$ with min $JL'[i].D$
        $i \leftarrow i + 1$
    for $l_i \leftarrow 0$ to $len(IL)$ do
        if $l_i \in JL$ and $l_i \notin AL$ then
            $AL \leftarrow$ append $l_i$
            $l_j \leftarrow JL[l_i]$
            $bp_j \leftarrow IL[l_i].bp + IL[l_j].bp$
            $cp_j \leftarrow len(IL[l_i].cp]) + len([IL[l_j].cp)$
            while True do
                $AL \leftarrow$ append $l_j$
                                                    ▷ Continue on the next page
```

Algorithm 2 BWP algorithm (Part 2)

```
                if $j_i \in JL$ and $l_i \notin AL$ then
                    $l_j \leftarrow JL[l_j]$
                    $bp_j \leftarrow append\ IL[l_j].bp$
                    $cp_j \leftarrow cp_j + len([IL[l_j].cp)$
                else
                    $ML \leftarrow$ append $\{bp : bp_j, cpl : cp_j\}$
                    break
                end if
        end if
    end for
    for $l_i \leftarrow 0$ to $len(IL)$ do
        if $l_i \notin AL$ then
            $ML \leftarrow$ append $\{bp : IL[l_j].bp, cpl : len(IL[l_j].cp)\}$
        end if
    end for
    return $ML$
end procedure
procedure Main($img, bin\_p, n, init, ws, mp_{max}$)
    $PL \leftarrow$ call $GetPotentialLines(img, bin\_p, n, ws, mpc_{max})$
    $IL \leftarrow$ call $RemoveRepeatedLines(PL)$
    $ML \leftarrow$ call $JoinLines(IL, {}^{mm}/_{px})$
    $FL \leftarrow n\ ML$ with $max(ML.cpl)$
    $RL \leftarrow fit(FL)$
    $L \leftarrow FL$ with min $|RL[init.x], init|$
    return $L$
end procedure
```

Table 3
DLO modeling problems detected by the self-critique and tuning module, and corrective actions.

Problem	Condition	Action
P1: The segmentation filter is too permissive	$success$ **and** $\frac{cp}{bin\_p} \cdot n < 0.6$ **and** $line\_lkh < 0.75$	**A1.1:** $thr_2 \cdot 0.75$ **A1.2:** $thr_1 \cdot 0.85$ **A1.3:** $thr_3 \cdot 0.85$ **A1.4:** $ws.x + {}^{px}/_{D} \cdot 0.5$ $ws.y + {}^{px}/_{D} \cdot 0.35$
P2: The segmentation filter too restrictive	**not** $success$ **and** (BWP **or** (**not** $init\_ok$ **or** $mp > mp_{max}$))	**A2.1:** $thr_2 \cdot 1.2$ **A2.2:** $thr_1 \cdot 1.2$ **A2.3:** $thr_3 \cdot 1.2$
P3: Occlusion or DLO entanglement	**not** $success$ **and** FWP **and** $init\_ok$ **and** $mp < mp_{max}$	**A3.1:** $ws.x + {}^{px}/_{D} \cdot 0.5$ $ws.y + {}^{px}/_{D} \cdot 0.35$ **A3.2:** $mpc_{max} + 1$ **A3.3:** $thr_2 \cdot 1.2$ $thr_1 \cdot 1.2$

factors on the system is very difficult to predict, which makes the shape estimation of the DLO very challenging. Furthermore, the selection of the correct system parameters when the conditions are not constant increases the complexity of the prediction process. To address this issue and to make the system more robust against unforeseen situations, the self-critique and tuning module has been introduced. This module evaluates the obtained results and, if they are not satisfactory, it performs a corrective action, fine-tuning the system parameters for the next iteration. The critiques can be applied with both the FWP and the BWP algorithms, referring to them as FWP-C and BWP-C respectively.

This module defines different conditions that are used to identify problems on the DLO estimation. Then, for each problem, the module executes several tuning actions, as can be seen in Table 3. To see their effect independently, just one tuning action is performed per iteration, and their indices indicate their priority. If after a second iteration the same problem is detected, the module executes the next corrective action and, when all of them are finished, it starts again from the beginning. In order to prevent long computation times, if the result is not successful after six iterations the last estimation of the DLO shape is returned.

As can be seen in Table 3, three possible DLO estimation problems are considered. The first issue is the use of a very permissive filter for the DLO segmentation. This problem is identified when the result of the points propagation is considered successful, but the percentage of segmented pixels captured by the evaluation windows ($\frac{cp}{bin\_p}$) is lower than 60% (divided by the number of DLOs of the evaluated color (n)) and the line likelihood ($line\_lkh$) is lower than 0.75. The line likelihood metric is calculated to evaluate if the estimation of the DLO shape could have been altered by the noise that passed the segmentation filter. It is defined as the ratio between the total number of segmented pixels inside evaluation windows of height equal to the DLO diameter located at the estimated DLO points, and the total number of segmented pixels inside windows of double this height at the same points. In a perfectly segmented DLO, with no noise, all the segmented pixels should be within the DLO diameter, so this factor would be equal to one. The detection of this circumstance (P1) means that the segmentation is detecting a lot of noise and, although the resultant DLO shape is not necessarily bad, there is no certainty that it is correct. To address this issue, the corrective actions are aimed at making the segmentation filter more restrictive by decreasing its threshold values. Moreover, a fourth action tunes the evaluation window size (ws).

Opposite to the first issue, the second problem (P2) appears when the segmentation filter is too restrictive, causing the loss of DLO information (i.e., the number of false negative pixels is high), which, in turn, makes it impossible to reconstruct the DLO's shape. This situation is more difficult to detect than the previous one, as the loss of information could be caused by occlusions or entanglements of the DLOs (i.e., P3) and not by an incorrect segmentation. The failure of the DLO points propagation is a common condition for detecting both problems (P2 and P3), but it is possible to differentiate them by evaluating the cause of this failure. When the FWP algorithm is employed, if the points propagation fails because the initial point of the DLO cannot be found or because the total number of evaluation windows that do not contain any segmented pixel (*mp*) is higher than the maximum, P2 is identified as the problem. This consideration is done because the results indicate that there are many discontinuities in the segmented DLO. Whereas, if the problem was caused by an occlusion (P3) there should be just one large gap and, in that case, the cause of failure would be that the number of consecutive evaluation windows without segmented pixels (*mpc*) is higher than the maximum. On the other hand, when the BWP algorithm is used, P2 is always identified as the reason for failure, as the algorithm is able to join the segments that belong to the same DLO (i.e., it is robust against occlusions).

Regarding the corrective actions of the previous two problems, if a highly restrictive segmentation is identified as the issue (P2), its threshold values are increased to make it more permissive. Conversely, if the cause of incorrect estimation is deemed to be an occlusion (P3), three different measures can be applied: increasing the evaluation window size, increasing the maximum number of consecutive evaluation windows without segmented pixels, and making the filter more permissive.

4. Experimental evaluation

The performance of the different modules of the system, as well as the performance of the system as a whole, has been tested with several images of a different number of deformable objects of different sizes and colors. In particular, the developed system was used to estimate the shape of the cables composing three different wiring harnesses. The first one (WH1) is composed by ten cables of six different colors with a diameter of 1.3 mm, the second one (WH2) counts with eleven cables of eight different colors and also 1.3 mm diameter, and the third one (WH3) is composed by six cables of five different colors and they have a bigger diameter, 2.1 mm. Moreover, the connector of the third wiring harness, from which their cables go out, has two rows, causing occlusions in the initial points of the cables of the lower layer. The three analyzed components are composed by relatively small cables, of different colors, crossing each other, and with small distances between them (around 1 mm between their starting points), which makes them very appropriate specimens to test the system robustness.

Fig. 6 shows the results of the analysis of an image of WH1 with the FWP algorithm for determining the DLO points. As can be seen in the top left image (a), this wiring harness is composed by two green cables, two yellow cables, two blue cables, two white cables, a red cable and a black cable. The white cables were excluded from the analysis as their color was very similar to the background and their diameter was very small, making its detection difficult even for the human sight. In the top center image (b), the system is used without the feedback provided by the self-critique tuning module (FWP). The shape of the green, yellow, red and black cables are determined correctly, even when the lightning is not constant and when there are entanglements and occlusions between them. However, the system fails for modeling the blue cables due to a bad segmentation, with highly restrictive thr_1 and thr_2 filter parameters, that cannot detect well the pixels of the initial points of the cable, as can be seen in the bottom center image (e). This problem is solved when using the self-critique module (FWP-C) that iterates optimizing the filter parameters, as can be seen in the bottom right image (f). The result is the image in the top right (c), where all the cables are modeled successfully. Additionally, the bottom left figure (d) shows another example of a bad segmentation of the blue cables, but, in this case, due to highly permissive filter parameters.

Table 4
Comparison of the accuracy and computation time of the different system algorithms at different image resolutions. The accuracy and its standard error (SE) are presented as percentages and the computation time is expressed as a multiple of the minimum time.

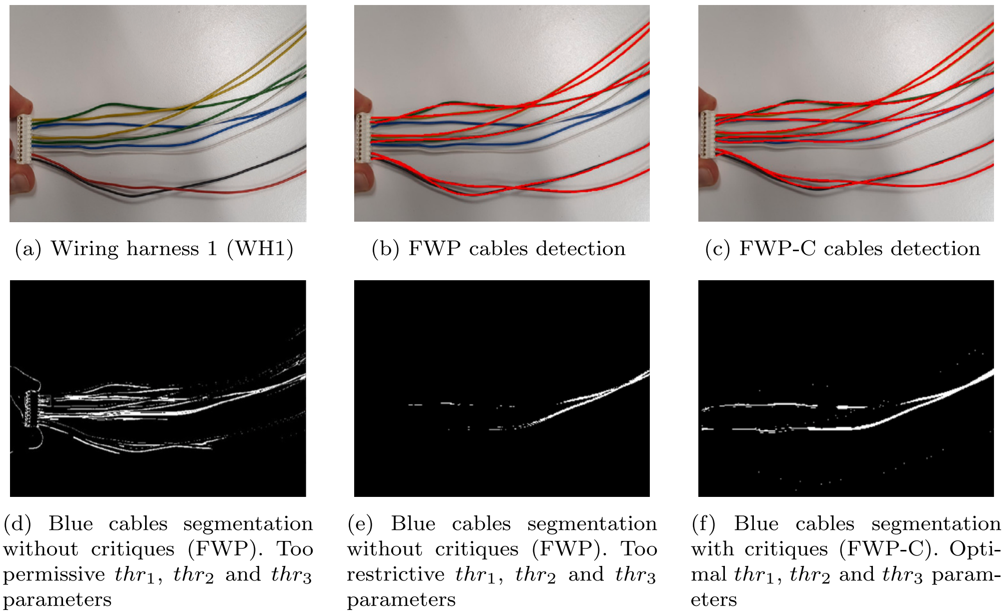
px/D	Parameter	FWP	FWP-C	BWP	BWP-C
2	Accuracy ± SE (%)	64 ± 8	65 ± 8	74 ± 9	77 ± 10
	Computation time	1	2.0	1.7	5.9
3	Accuracy ± SE (%)	64 ± 8	69 ± 8	75 ± 9	80 ± 9
	Computation time	1.7	3.7	3.1	8.4
4	Accuracy ± SE (%)	73 ± 3	77 ± 3	83 ± 3	91 ± 2
	Computation time	3.1	7.1	6.7	12.2

The ability of the system to detect adjacent tiny cables is tested with WH2, which has three branches of adjacent cables. Fig. 7 shows the results of analyzing an image of this wiring harness using the FWP-C algorithm. As can be seen in figures (b) and (c) ten out of eleven cables are detected successfully, even with large self-occlusions.

The effect of occlusions in the initial points of the DLOs on the FWP and BWP algorithms was tested with WH3 which, as described previously, has two rows of cables, causing occlusions at the beginning of the cables of the bottom layer. The results are shown in Fig. 8 and, as expected, the FWP algorithm fails for calculating some of the bottom cables, as it relies strongly on the cables initial point. This can be seen in the center image (b), where even using the self-critique tuning module (FWP-C), two of the bottom cables (gray and green) are detected wrong. In order to distinguish them, the estimated cable shape of the bottom cables is shown in green and the top cables in red. However, in the right image (c), the BWP-C algorithm was used, obtaining successful results for all the cables, as it analyzes all the potential DLOs of the image, without considering any fixed starting point. This figure demonstrates also the robustness against adjacent cables and self-occlusions of the two DLO points detection algorithms presented in this paper (FWP and BWP).

The performance of the system was also evaluated under challenging conditions, such as complex backgrounds and occlusions, as depicted in Fig. 9. The experimental results demonstrated that the four algorithms performed well, with a slight improvement when using the tuning module, particularly in scenarios where objects in the background could be confused with the cables. Regarding occlusions, the BWP algorithm works much better than the FWP one, as can be seen in the last row of Fig. 9, detecting all the cables successfully. The system also shows good performance for detecting DLOs whose color is very similar to the background color. This is analyzed in detail in Fig. 10, where all the steps followed for detecting a green cable on a green complex background using the BWP algorithm are presented. The color filter (b) alone is not good for segmenting the DLO as both the background and the cable are green. The edge detector (c) alone is not good either as the background is complex and many edges are detected. The segmentation improves significantly when merging the results of the two previous operators (d), reducing the noise substantially. Then, the BWP algorithm determines all the possible DLOs in the segmented image (e), and finally, it selects the most probable DLO candidate (f). The last two images (g, h) show the results and, as can be seen, the detection of the green cable is successful.

Table 4 presents a summary of the results obtained from the analysis of several images of the three evaluated wiring harnesses in different conditions, with both homogeneous and complex backgrounds, and in the presence and absence of occlusions. The table compares the average accuracy achieved and the average computation time spent by using the different algorithms of the system, as well as different image


(a) Wiring harness 1 (WH1)

(b) FWP cables detection

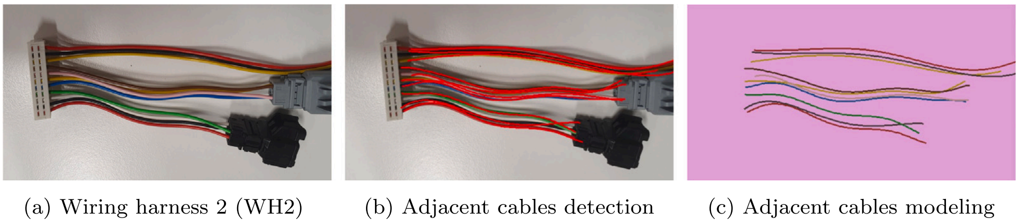
(c) FWP-C cables detection

(d) Blue cables segmentation without critiques (FWP). Too permissive thr_1, thr_2 and thr_3 parameters

(e) Blue cables segmentation without critiques (FWP). Too restrictive thr_1, thr_2 and thr_3 parameters

(f) Blue cables segmentation with critiques (FWP-C). Optimal thr_1, thr_2 and thr_3 parameters

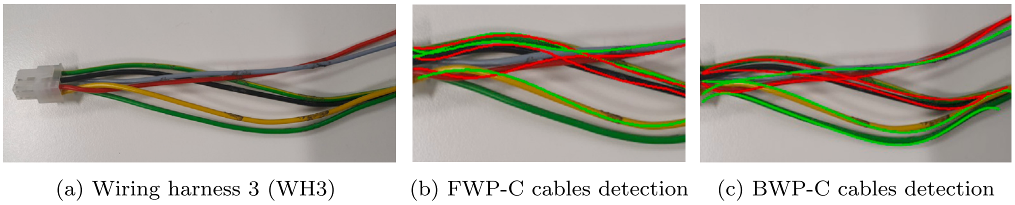
Fig. 6. FWP and FWP-C comparison for the shape estimation of the cables of WH1.


(a) Wiring harness 2 (WH2)

(b) Adjacent cables detection

(c) Adjacent cables modeling

Fig. 7. Shape estimation of the cables of WH2.


(a) Wiring harness 3 (WH3)

(b) FWP-C cables detection

(c) BWP-C cables detection

Fig. 8. Shape estimation of the cables of WH3. The bottom layer cables are modeled in green and the top ones in red.

resolutions. The accuracy is defined as the mean percentage of cables in each image for which the shape estimation is correct. Due to the lack of ground truth, the evaluation of the estimated DLO shapes was done visually. To make this evaluation systematic, a strict measure was used, considering the estimation successful only if it follows the cable shape along its entire length accurately. The computation times are expressed as a multiple of the fastest algorithm, i.e., FWP with 2 px/D, emphasizing the algorithms' comparison. However, in order to have also an absolute reference, the computation time of all the analyzed images, each of them containing six to eleven cables, was between 1 s and 1 minute. Regarding the image resolution, it is expressed as the number of pixels of the smaller cable diameter (px/D).

These results show that the accuracy of the system improves in all the cases when using the feedback from the self-critique tuning module, however, more iterations make the system slower. It can be also seen that BWP presents better results than FWP, mainly due to its better performance in the presence of occlusions (in the initial points or in any part of the cables), but the computation time increases as the initial cable points are unknown and more pixels need to be analyzed. Regarding the resolution, as expected, when it increases the accuracy increases, however, the time increases severely. Therefore, the selection of the resolution and the algorithm will depend on the requirements of the specific application, making a balance between speed and accuracy, and the characteristics of the input images (the type of background, the presence of occlusions, the DLOs size and distribution, etc.).

Finally, Fig. 11 shows a qualitative comparison of the developed system against two of the most advanced DLO identification models currently available in the state-of-the-art literature. Section 2 presents these two approaches, which employ different strategies: a CNN-based DLO segmentation was used in RT-DLO [20] and a superpixel segmentation in Ariadne [9]. These two identification models target the same problem as the one our model tries to solve, but from a different viewpoint. RT-DLO and Ariadne emphasize the fast identification of DLOs with arbitrary shapes, while our model targets the robust detection of multiple DLOs with minimum distance among those and small sections, the typical scenario faced in wiring harnesses. While all three approaches perform excellently in identifying medium-sized separated cables, our method outperforms the others when the evaluated DLOs are adjacent, very thin, and have constant overlapping.

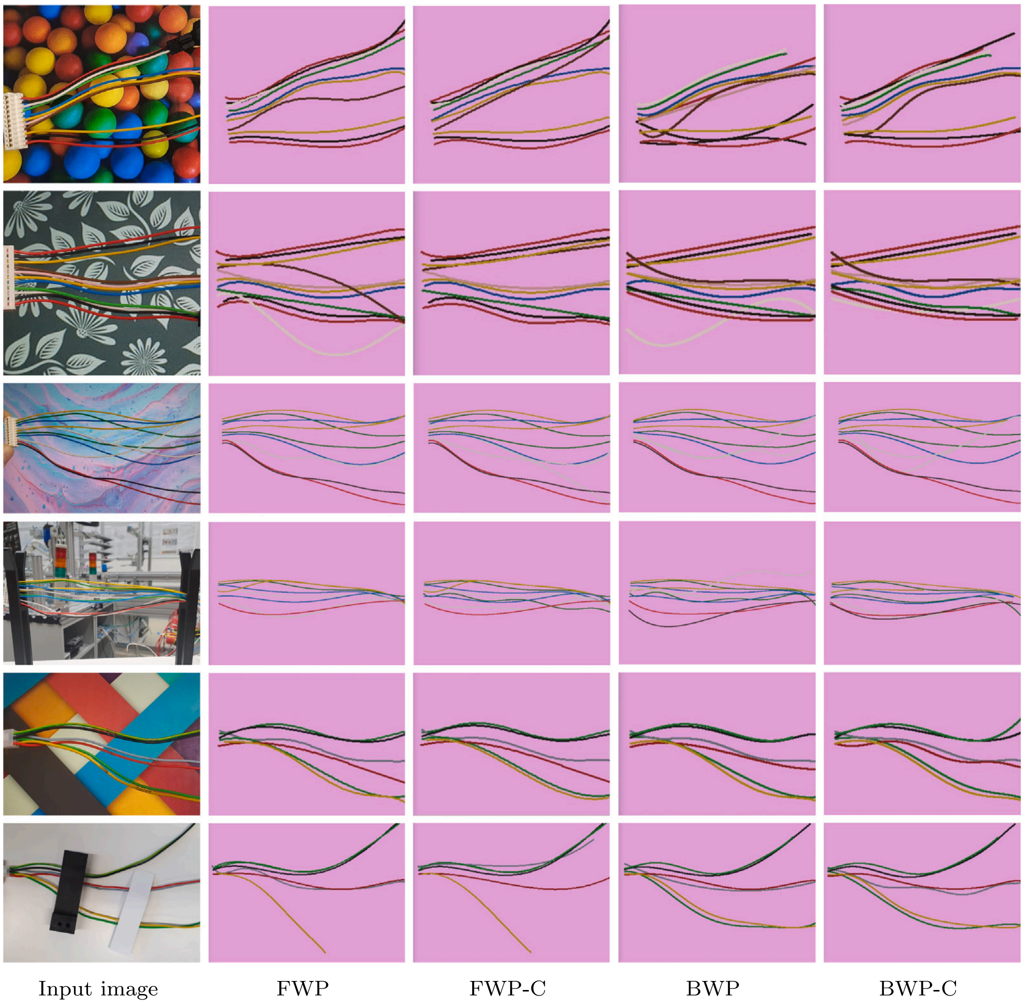
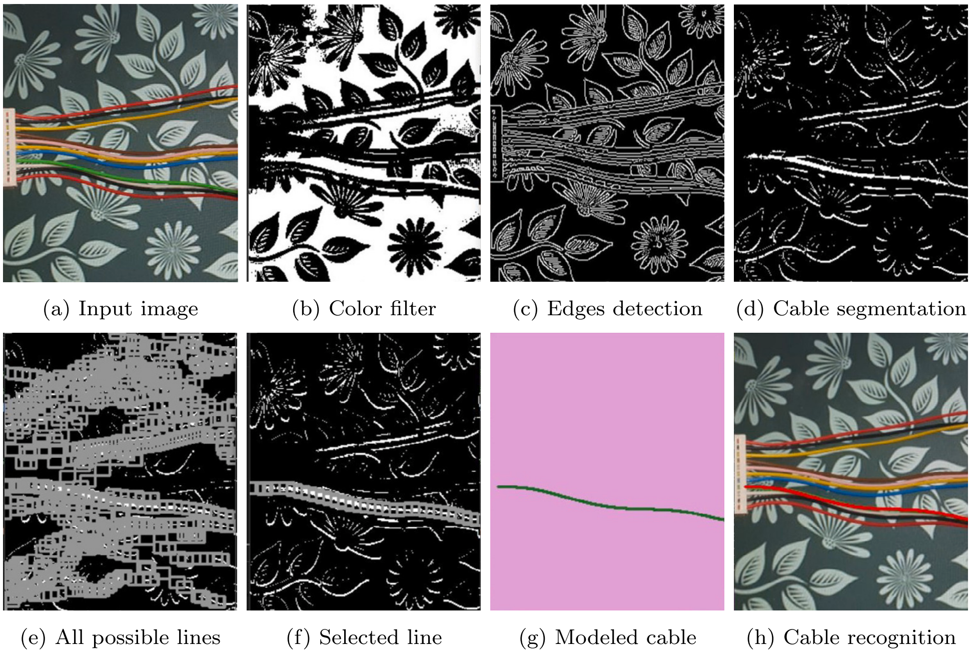


Fig. 9. Analysis of DLOs images with occlusions and complex backgrounds with the four algorithms of the system using 4 $^{px}/_{D}$.



(a) Input image (b) Color filter (c) Edges detection (d) Cable segmentation

(e) All possible lines (f) Selected line (g) Modeled cable (h) Cable recognition

Fig. 10. BWP DLO detection process steps for recognizing the green cable.

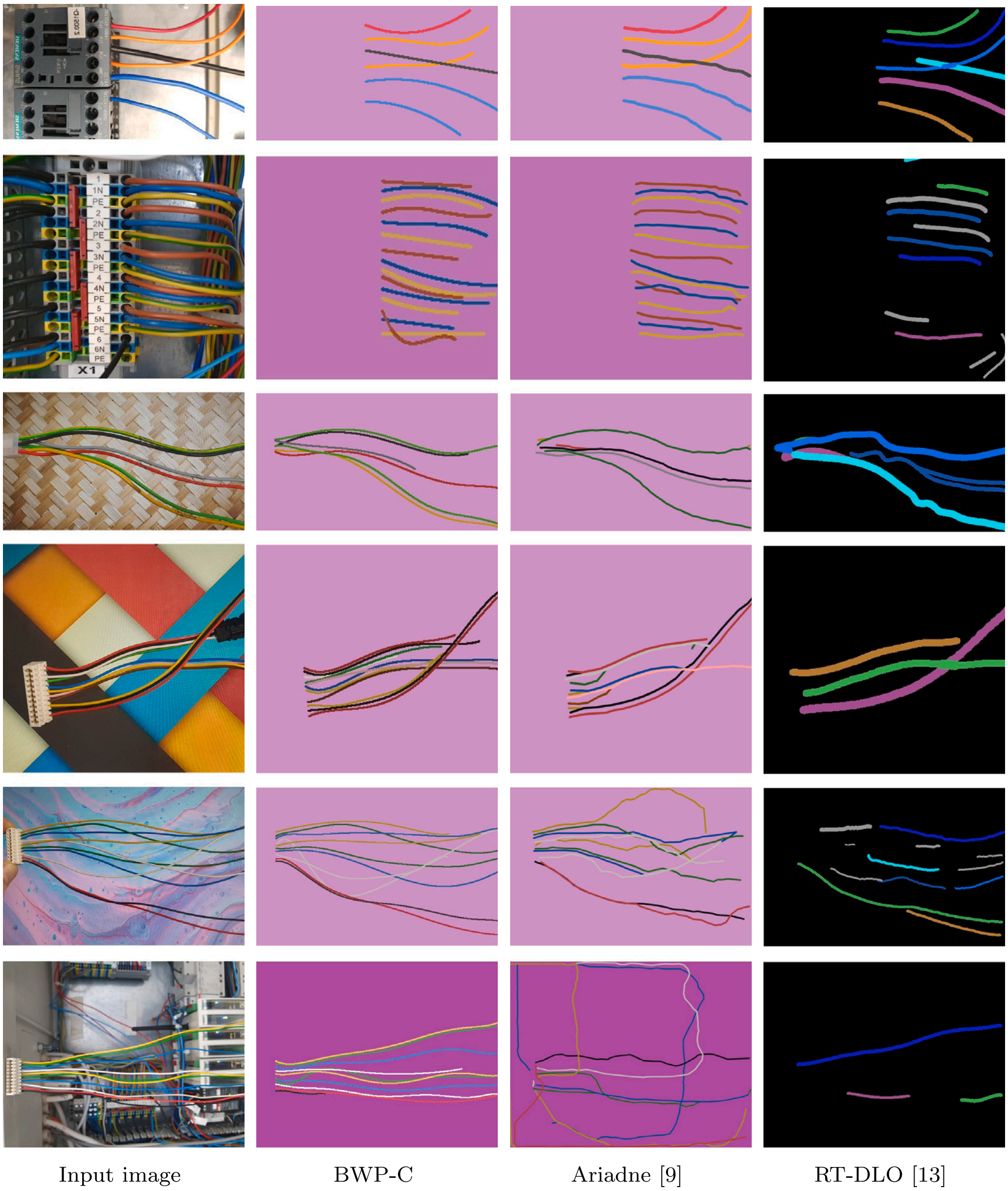


Fig. 11. Qualitative comparison of BWP-C with other state-of-the-art algorithms.

Table 5
Characteristics and capabilities of the developed vision-based DLO detection system. Ent.: Entanglements, Occ.: Occlusions, Adj.: Adjacent, NT: Not tested.

Test specimens	**DLOs number**	**Size (D/L)**	**DLOs color**	**Background**	**Ent.**	**Occ.**	**Adj.**
Wiring harness	Multiple	Tiny	Multiple	Complex	✓	✓	✓
Approach		**Prev info**		**Outcome**	**Camera**		**Speed**
DLO segmentation + points propagation (FWP or BWP) + regression + unsupervised critique		Init points, DLO colors and diameters		Polynomial lines	2D RGB		< 1 min

Moreover, the main characteristics and capabilities of the system are summarized in Table 5, which can be used to compare it with the rest of approaches reviewed in the paper (Tables 1 and 2). As the table shows, the main strengths of the proposed approach are its ability to identify the shape of multiple adjacent thin DLOs of different colors, and its robustness against entanglements of the DLOs, occlusions, and complex backgrounds. However, its main drawback is its computation time, not being adequate for real-time tracking applications.

5. Conclusions

The automatic manipulation of deformable materials is a challenging task, where perception plays a key role, as the shape of these objects is not constant and is very difficult to predict. In this paper, a novel vision-based system for the detection and shape estimation of DLOs is presented. This system includes a DLO segmentation algorithm that merges color and edge information to reduce the noise, two novel algorithms for detecting the DLO points, and an unsupervised critique module that fine-tunes the system parameters. The system has been tested with images of three wiring harnesses in different conditions, demonstrating its ability to detect multiple adjacent tiny DLOs of different colors in homogeneous and complex backgrounds, and with entanglements between them. Additionally, the BWP algorithm has been proved to be robust against occlusions, both in the initial points and in any part of the DLOs.

The different operation modes of the system were compared during the tests showing a systematic improvement in the performance when using the critique module and the background propagation, reaching an average DLO recognition of 91% with the BWP-C algorithm and a resolution of 4 pixels per DLO diameter. However, all the changes intended to improve the system performance, bring also an increase in the computation time, therefore, there is not a perfect configuration of resolution and algorithm, and the selection will depend on the application requirements.

Despite the good results obtained by the developed system, it is important to highlight its limitations in order to provide a more comprehensive view of its capabilities and potential applications. The system models the DLOs using polynomial functions, which provide a mathematical representation of their shape. This is useful in many applications, for instance, to determine the maximum distance between DLOs, however, it restricts its use to DLOs without self-intersections and significant changes in direction. Additionally, the system is not adequate for tracking applications as it cannot operate in real time.

While acknowledging these limitations, it should be noted that they fall outside the intended scope of the developed approach and, therefore, addressing them is not one of the objectives of our future research. Instead, the focus will be on enhancing the system's performance, in particular by increasing its accuracy and reducing its computation time. Furthermore, our future work will aim to expand the system by developing new functionalities to support robots in DLO manipulation tasks, such as the determination of the optimal grasp points or the verification of the correct position and configuration of the DLOs after the manipulation.

CRediT authorship contribution statement

Pablo Malvido Fresnillo: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration. **Saigopal Vasudevan:** Formal analysis, Writing – original draft, Writing – review & editing. **Wael M. Mohammed:** Formal analysis, Writing – review & editing. **Jose L. Martinez Lastra:** Conceptualization, Supervision, Funding acquisition. **Jose A. Perez Garcia:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pablo Malvido Fresnillo reports financial support was provided by Horizon 2020. Saigopal Vasudevan reports financial support was provided by Horizon 2020. Jose L.Martinez Lastra reports financial support was provided by Horizon 2020.

Data availability

Data will be made available on request.

Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement n° 870133, correspondent to the project entitled REMODEL, Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects. All authors approved the final version of manuscript to be published.

References

[1] Khalifa A, Palli G. New model-based manipulation technique for reshaping deformable linear objects. Int J Adv Manuf Technol 2022;118(11–12):3575–83.

[2] Zhu J, Cherubini A, Dune C, Navarro-Alarcon D, Alambeigi F, Berenson D, et al. Challenges and outlook in robotic manipulation of deformable objects. IEEE Robot Autom Mag 2022;2–12. http://dx.doi.org/10.1109/MRA.2022.3147415.

[3] Sanchez J, Corrales J-A, Bouzgarrou B-C, Mezouar Y. Robotic manipulation and sensing of deformable objects in domestic and industrial applications: A survey. Int J Robot Res 2018;37(7):688–716, Publisher: SAGE Publications Ltd STM.

[4] Yin H, Varava A, Kragic D. Modeling, learning, perception, and control methods for deformable object manipulation. Science Robotics 2021;6(54):eabd8803. http://dx.doi.org/10.1126/scirobotics.abd8803.

[5] Fresnillo PM, Vasudevan S, Mohammed WM, Martinez Lastra JL, Laudante G, Pirozzi S, et al. Deformable objects grasping and shape detection with tactile fingers and industrial grippers. In: 2021 4th IEEE international conference on industrial cyber-physical systems. 2021, p. 525–30. http://dx.doi.org/10.1109/ICPS49255.2021.9468151.

[6] Caldwell TM, Coleman D, Correll N. Optimal parameter identification for discrete mechanical systems with application to flexible object manipulation. In: 2014 IEEE/RSJ international conference on intelligent robots and systems. 2014, p. 898–905. http://dx.doi.org/10.1109/IROS.2014.6942666.

[7] Cirillo A, Laudante G, Pirozzi S. Proximity sensor for thin wire recognition and manipulation. Machines 2021;9(9):188. http://dx.doi.org/10.3390/machines9090188, Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.

[8] Tang T, Wang C, Tomizuka M. A framework for manipulating deformable linear objects by coherent point drift. IEEE Robot Autom Lett 2018;3(4):3426–33. http://dx.doi.org/10.1109/LRA.2018.2852770.

[9] De Gregorio D, Palli G, Di Stefano L. Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation. In: Computer vision – ACCV 2018. Cham: Springer International Publishing; 2019, p. 662–77. http://dx.doi.org/10.1007/978-3-030-20890-5_42.

[10] Javdani S, Tandon S, Tang J, O'Brien JF, Abbeel P. Modeling and perception of deformable one-dimensional objects. In: 2011 IEEE international conference on robotics and automation. 2011, p. 1607–14. http://dx.doi.org/10.1109/ICRA.2011.5980431.

[11] Sie Ho Lee T, Fidler S, Dickinson S. Detecting curved symmetric parts using a deformable disc model. In: Proceedings of the IEEE international conference on computer vision. 2013, p. 1753–60.

[12] Wang Y, McConachie D, Berenson D. Tracking partially-occluded deformable objects while enforcing geometric constraints. In: 2021 IEEE international conference on robotics and automation. 2021, p. 14199–205.

[13] Abegg F, Henrich D, Wörn H. Manipulating deformable linear objects - vision-based recognition of contact state transitions -. Technische Universitat Kaiserslautern; 1999.

[14] Huo S, Duan A, Li C, Zhou P, Ma W, Wang H, et al. Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework. IEEE Robot Autom Lett 2022;7(2):5222–9.

[15] Borum A, Matthews D, Bretl T. State estimation and tracking of deforming planar elastic rods. In: 2014 IEEE international conference on robotics and automation. 2014, p. 4127–32.

[16] He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision. 2017, p. 2961–9.

[17] Nguyen HG, Franke J. Deep learning-based optical inspection of rigid and deformable linear objects in wiring harnesses. Procedia CIRP 2021;104:1765–70. http://dx.doi.org/10.1016/j.procir.2021.11.297.

[18] Song Y, Yang K, Jiang X, Liu Y. Vision based topological state recognition for deformable linear object untangling conducted in unknown background. In: 2019 IEEE international conference on robotics and biomimetics. 2019, p. 790–5. http://dx.doi.org/10.1109/ROBIO49542.2019.8961652.

[19] Dai Z, Yi J, Zhang Y, Zhou B, He L. Fast and accurate cable detection using CNN. Appl Intell 2020;50(12):4688–707.

[20] Caporali A, Galassi K, Žagar BL, Zanella R, Palli G, Knoll AC. RT-DLO: Real-time deformable linear objects instance segmentation. IEEE Trans Ind Inf 2023;1–10. http://dx.doi.org/10.1109/TII.2023.3245641.

[21] Redmon J, Farhadi A. YOLOv3: An incremental improvement. 2018, arXiv preprint arXiv:1804.02767.

[22] Foresti G, Pellegrino F. Automatic visual recognition of deformable objects for grasping and manipulation. IEEE Trans Syst Man Cybern Part C (Appl Rev) 2004;34(3):325–33. http://dx.doi.org/10.1109/TSMCC.2003.819701.

[23] Matsuno T, Tamaki D, Arai F, Fukuda T. Manipulation of deformable linear objects using knot invariants to classify the object condition based on image sensor information. IEEE/ASME Trans Mechatronics 2006;11(4):401–8.

[24] Keipour A, Mousaei M, Bandari M, Schaal S, Scherer S. Detection and physical interaction with deformable linear objects. 2023, http://dx.doi.org/10.48550/arXiv.2205.08041, arXiv preprint arXiv:2205.08041.

[25] Wnuk M, Hinze C, Zürn M, Pan Q, Lechler A, Verl A. Tracking branched deformable linear objects with structure preserved registration by branch-wise probability modification. In: 2021 27th international conference on mechatronics and machine vision in practice. 2021, p. 101–8.

[26] Lui WH, Saxena A. Tangled: Learning to untangle ropes with RGB-D perception. In: 2013 IEEE/RSJ international conference on intelligent robots and systems. 2013, p. 837–44. http://dx.doi.org/10.1109/IROS.2013.6696448.

[27] Wnuk M, Hinze C, Lechler A, Verl A. Kinematic multibody model generation of deformable linear objects from point clouds. In: 2020 IEEE/RSJ international conference on intelligent robots and systems. 2020, p. 9545–52.

[28] Fatan M, Daliri mr, Mohammad Shahri A. Underwater cable detection in the images using edge classification based on texture information. Measurement 2016;91. http://dx.doi.org/10.1016/j.measurement.2016.05.030.

[29] Kuhn VN, Drews PLJ, Gomes SCP, Cunha MAB, Botelho SSdC. Automatic control of a ROV for inspection of underwater structures using a low-cost sensing. J Braz Soc Mech Sci Eng 2015;37(1):361–74. http://dx.doi.org/10.1007/s40430-014-0153-z.

[30] Nakagaki H, Kitagi K, Ogasawara T, Tsukune H. Study of insertion task of a flexible wire into a hole by using visual tracking observed by stereo vision. In: Proceedings of IEEE international conference on robotics and automation. Vol. 4. 1996, p. 3209–14.

[31] Greminger MA, Nelson BJ. A deformable object tracking algorithm based on the boundary element method that is robust to occlusions and spurious edges. Int J Comput Vis 2008;78(1):29–45.

[32] Fletcher R. Practical methods of optimization. Chichester; New York: Wiley; 1987.

[33] Felzenszwalb P. Representation and detection of deformable shapes. IEEE Trans Pattern Anal Mach Intell 2005;27(2):208–20.

[34] Abegg F, Engel D, Worn H. A robust algorithm for segmenting deformable linear objects from video image sequences. In: Proceedings 15th international conference on pattern recognition. Vol. 4. ICPR-2000, 2000, p. 756–9. http://dx.doi.org/10.1109/ICPR.2000.903027.

[35] Ortiz A, Simó M, Oliver G. A vision system for an underwater cable tracker. Mach Vis Appl 2002;13(3):129–40. http://dx.doi.org/10.1007/s001380100065.

[36] Frangi AF, Niessen WJ, Vincken KL, Viergever MA. Multiscale vessel enhancement filtering. In: Medical image computing and computer-assisted intervention. Berlin, Heidelberg: Springer; 1998, p. 130–7.

[37] Staal J, Abramoff M, Niemeijer M, Viergever M, van Ginneken B. Ridge-based vessel segmentation in color images of the retina. IEEE Trans Med Imaging 2004;23(4):501–9.

[38] Pătrăucean V, Gurdjos P, von Gioi RG. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In: Computer vision. Vol. 7573. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012, p. 572–85. http://dx.doi.org/10.1007/978-3-642-33709-3_41.

[39] Geiger D, Gupta A, Costa L, Vlontzos J. Dynamic programming for detecting, tracking, and matching deformable contours. IEEE Trans Pattern Anal Mach Intell 1995;17(3):294–302.

[40] Zhou RW, Quek C, Ng GS. A novel single-pass thinning algorithm and an effective set of performance criteria. Pattern Recognit Lett 1995;16(12):1267–75. http://dx.doi.org/10.1016/0167-8655(95)00078-X.

Pablo Malvido Fresnillo is a Doctoral Researcher at Tampere University (Finland). He received a M.Sc. in Electronics and Automation Engineering from University of Vigo (Spain) in 2019 and a B.Sc. in Industrial Engineering from University of Vigo (Spain) in 2017. As he is pursuing a Doctoral Degree in Engineering Sciences, Mr. Malvido's research interests include Robotics, Programming by Demonstration, Robot Bimanual Manipulation, Knowledge Based Reasoning Engines and Factory Automation. In addition, Mr. Malvido is part of the FAST-Lab research group and, since 2020, he is working in the project REMODEL, part of the EU H2020 funding program, whose aim is the development and implementation of hardware and software technologies to manipulate complex Deformable Linear Objects (DLOs) using robotic manipulators.

Saigopal Vasudevan is a Project Researcher for the FAST-Lab research group, at Tampere University. He received a M.Sc. in Factory Automation and Industrial Engineering from Tampere University, Finland in 2019 and a B.E (bachelor's degree) in Mechanical Engineering from Anna University, India in 2016. Mr. Vasudevan has worked in the automation industry as a Roboticist for a year, between 2016 and 2017. He is currently working in the project REMODEL from the EU H2020 funding program, since 2019; where the aim is to develop hardware and software technologies to manipulate complex Deformable Linear Objects (DLOs) using robotic manipulators. And since 2020, he is also working in the project AISOLA of the INSO profiling action, which is funded by the academy of Finland; whose objective is to assess social isolation and to prevent the onset of perceived loneliness in older adults, with a focus on developing and utilizing dedicated AI systems. In addition to working as a researcher, he also involved in collaborating for proposals, representing Tampere University, for the Horizon funding program of the EU with considerable success. Mr. Vasudevan's interests include Robotics, Bimanual Robot Manipulation, AI systems and Industrial Informatics & Automation.

Wael M. Mohammed is a Doctoral Researcher at Tampere University. He received a M.Sc. in Automation Engineering from Tampere University of Technology in 2017 and a B.Sc. in Mechatronics Engineering from university of Jordan in 2010. As he is pursuing a Doctoral Degree in Engineering Sciences, Mr. Mohammed's research interests include Robotics, Digital Twins, Knowledge Based Reasoning Engines and Factory Automation. In addition, Mr. Mohammed has been involved in writing proposals for research and innovation project funded by the EU commission. In 2010, he worked as a research assistant in the Production Engineering Department at Tampere University of Technology. Then in 2011, Mr. Mohammed worked as a head of the technical department in the Traffic Management System project at Etihad Alafandi L.L.C. in Saudi Arabia.

Prof. Lastra joined Tampere University of Technology in 1997 and became University Full Professor in 2006. His research interest is in applying Information and Communication Technologies to the field of Automation. He leads the FAST-Lab. with the ultimate goal of seamlessly integrating the knowledge of humans and machines in order to create smart environments. Prof. Lastra has co/authored over 300 articles and holds a number of patents in the field of Industrial Informatics and Automation. Prof. Lastra is an active member within the international research scene with a track record of 21 funded European Research projects. He serves as Associate Editor of the IEEE Transactions on Industrial Informatics, and He was a Technical Editor of the IEEE/ASME Transactions on Mechatronics in 2015–2016. Prof. Lastra served as Guest Editor for the MDPI Sensors Journal, and is an editorial member of the MDPI Machines journal. Furthermore, Prof. Lastra served as Technical Secretary for the IEEE Technical Committee on Industrial Cyber-Physical Systems in 2017–2018. For the period 2019-2022, He serves as a Co-Chair for the IEEE Technical Committee on Industrial Cyber-Physical systems.

Dr. José A. Pérez García worked as a production engineer for 6 years before he joined University of Vigo (Spain) in 1998, where he became Associate Professor in 2009. His research interest is on CAD/CAM and manufacturing technologies, fields in which he has developed a number of collaborations with several spanish companies. Dr. Perez has taught CAM courses in several Universities, both in Europe and America, and has co-authored around 40 scientific articles and conference papers. He holds a M.Sc. in Industrial Engineering, a M.Sc. in Labour Risk Prevention and a Ph.D. in Design & Manufacturing, all of them from University of Vigo.