IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

CVF-DLO: Cross-Visual-Field Branched Deformable Linear Objects Route Estimation

Chang Yu[1], Jianjian Wang[1], Pingfa Feng[1], Dingwen Yu[1], and Jianfu Zhang[1,†]

***Abstract*—The perception of deformable linear objects (DLOs) poses significant challenges in robotic manipulation. Crossovers, mergings, and bifurcations of multiple DLOs complicate the identification of individual DLO physical instances. Furthermore, DLOs are often too large to be captured by a single camera, requiring the stitching of multiple overlapping views. This paper presents *CVF-DLO*, a cross-visual-field route estimation framework of branched DLOs (BDLOs) laid along physical surfaces such as wire harnesses, based on images from multiple viewpoints and pose-aware cameras. *CVF-DLO* is applicable to various perception tasks involving DLO-like structures, such as verifying connection accuracy and route consistency in cables and pipes. We propose a DLO instance segmentation method that demonstrates superior performance in handling crossings and bifurcations. The extracted DLO paths are projected onto the designed cable-laying surfaces using the camera pose and scene model. Finally, DLO routes are retrieved by searching within the spatial path domain formed by intersecting visual fields. To validate our method on wiring harnesses and intersections, we use two public DLO datasets and introduce a new BDLO dataset to benchmark against state-of-the-art DLO instance segmentation methods. Additionally, we present a cabin wiring harness dataset to evaluate the performance of the cross-visual-field route estimation. We have released all our source code, supplementary video, and datasets (with ground truth) at https://github.com/ForNe-tech/CVF-DLO.**

***Index Terms*—deformable linear objects, instance segmentation, route estimation, computer vision, perception for manipulation**

I. Introduction

Flexible objects, such as cables, hoses, and wire harnesses, require significant manual effort for both manipulation and perception [1], [2]. These objects are commonly referred to as deformable linear objects (DLOs). Modeling, perceiving, and manipulating DLOs are considerably more complex than for rigid objects [3]. Their flexibility, uniform properties, and tendency to entangle pose significant challenges for instance segmentation. This study investigates DLO route estimation using multiple perspectives. The subtle textures, limited feature representation, and irregular geometry of DLOs challenge conventional methods. Segmenting branched DLOs requires handling complex intersections and bifurcations. Additionally, estimating complete DLO routes across visual fields remains a nontrivial and underexplored problem.

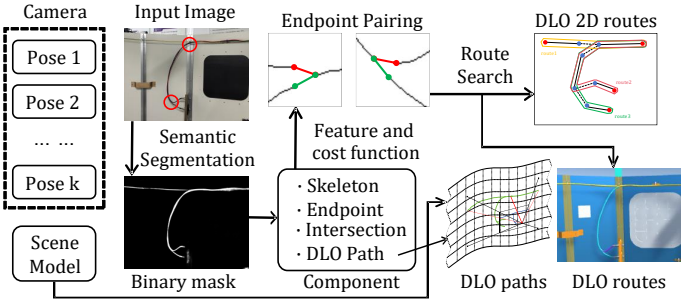


Fig. 1. Overview of *CVF-DLO* pipeline.

This study introduces *CVF-DLO* (Cross-Visual-Field DLO Route Estimation), a method for estimating the routes of branched deformable linear objects along physical surfaces from multiple perspectives. Our method takes a color image as input and generates a binary DLO mask using a deep convolutional neural network (DCNN). The skeleton is extracted from the mask and segmented into distinct paths, including intersections and path endpoints. Route ambiguity caused by intersections is formulated as the problem of accurately connecting path endpoints. To address this, we propose an endpoint pairing cost function. Using the camera pose and scene model, the pixel paths are projected onto predefined cable-laying surfaces to estimate their spatial geometry. A DLO route search algorithm then identifies potential routes within the domain of spatial DLO paths and connects the paths sequentially to reconstruct complete DLO routes. An overview of the *CVF-DLO* pipeline is shown in Fig. 1.

Our method processes DLO instance segmentation at over 15 frames per second (FPS) with an input resolution of 640 × 360 pixels on a desktop equipped with a 2.90 GHz Intel® Core™ i7-10700 CPU and 16 GB RAM. For real-time camera pose estimation, DLO images are captured using the Hololens2, which features an Intel Atom x5-Z8100 CPU and 2 GB RAM.

Overall, our main contributions in this article are that

1) We develop an endpoint pairing cost function and a connection strategy to resolve DLO bifurcations, crossings, and multi-DLO intersections.

2) We propose a method for reconstructing spatial DLO

Manuscript received: February 26, 2025; Accepted: June 18, 2025. This paper was recommended for publication by Associate Editor M. Vincze upon evaluation of the Associate Editor and Reviewers' comments. This research was supported by a grant from the National Natural Science Foundation of China (No. 52475522).

[1]Chang Yu, Jianjian Wang, Pingfa Feng, Dingwen Yu, and Jianfu Zhang are with the State Key Laboratory of Tribology in Advanced Equipment, Department of Mechanical Engineering, Tsinghua University, Beijing 10084, China (email: yc2016thu@163.com; wangjjthu@tsinghua.edu.cn; fengpf@tsinghua.edu.cn; yudw@tsinghua.edu.cn; zhjf@tsinghua.edu.cn).

† Corresponding author.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

paths and integrating them across multiple visual fields.

3) We introduce a route search algorithm for branched deformable linear objects (BDLOs) to address route ambiguity caused by bifurcations, branches, and knots. Combined with Contribution 1, it forms a pipeline for efficient and accurate DLO instance segmentation.

4) We present a pipeline for cross-visual-field BDLO route estimation, enabling the reconstruction of complete DLO route networks within a scene.

The remainder of this paper is organized as follows. Section II reviews related work. Section III details our cross-visual-field BDLO route estimation method. Section IV presents experimental results and comparisons with state-of-the-art approaches. Finally, Section V concludes the paper and discusses future research directions.

II. Related Work

This study reviews existing literature from three perspectives: DLO semantic segmentation, DLO instance segmentation, and DLO 3D shape estimation.

DLO semantic segmentation aims to distinguish DLO pixels from the background. Early methods primarily relied on color, contrast, or markers [4], [5], [6], [7], but these approaches assume that the background exhibits a clear contrast with the DLOs. For instance, [6] abstracts DLOs into several connected chains, while *Ariadne* [8] uses DLO superpixels for image segmentation. However, these methods are less robust compared to data-driven, deep learning-based approaches. [9] employs a deep learning-based optical inspection method to segment rigid components and DLOs in wiring harnesses.

A major challenge with data-driven methods for DLO segmentation is the difficulty in obtaining high-quality DLO labels. To address this issue, several pipelines have been developed to generate DLO image samples using synthetic methods [10], [11], [12], [13]. [14] introduces the Electric Wires Dataset, a public DLO synthesis dataset. This dataset is used to train *Deeplabv3+* [15], achieving promising results in DLO pixel segmentation.

Ariadne[8] is one of the earliest perceptual algorithms for segmenting DLO instances. It introduces the "walks" algorithm, which searches for DLO paths by traversing superpixels belonging to DLOs. The algorithm addresses path ambiguity caused by intersections through color, distance, and curvature. *Ariadne+* [16] enhances this method by utilizing a deep convolutional neural network (DCNN) to extract binary masks of DLOs, significantly improving computational efficiency. A neural network is also employed to determine the most probable paths at intersections. *FASTDLO* [17] represents DLOs using skeleton pixels, increasing processing speed from 3 FPS in *Ariadne+* to 20 FPS. *RT-DLO* [18] further advances real-time performance by performing sparse sampling on skeleton pixel curves, offering advantages in handling DLO mask discontinuities but also increasing the likelihood of segmentation errors. *mBEST* [19] is regarded as one of the most advanced DLO perception algorithms. By incorporating cumulative bending energy minimization to handle intersections, mBEST effectively addresses DLO instance segmentation challenges involving multiple crossings, entanglements, and self-crossings.

The above methods are designed for scenes with unbranched DLOs. In BDLOs, multiple DLOs are bundled together with branches. [20] attempts to detect a strand of BDLOs as a single target. [21] extracts topological features such as connectors, clamps, branch points, and overlaps on BDLOs. [22] proposes a topological matching method based on undirected graphs to match DLO segments with varying shapes across views. However, [22] assumes the image contains only one BDLO without intersections.

Compared to planar DLO segmentation, studies on 3D shape estimation of DLOs are limited. [23] estimates cable shapes using force/torque sensor measurements. [24] and [25] estimate DLO 3D shapes through unobscured point clouds. [26] groups and refines DLO point clouds using KD-Tree to calculate cable parameters. However, due to the small volume of DLOs, high resolution is required for point clouds. It is easier to obtain DLO pixels from 2D images and project them into 3D space. [27] proposes a method to estimate DLO spatial position from images, assuming a limited operating plane. Further, [28] uses the projection relationship between equal-width cables and DLO edges in images to estimate depth and 3D shape. *DLO3DS* [29] and [30] use a multi-view approach to observe the same DLO section from different perspectives, determining DLO points in space based on the intersection of multiple sightlines. However, these methods require highly precise camera poses, such as from robot-mounted cameras, or high-resolution depth information.

CVF-DLO, proposed in this paper, projects extracted DLO pixels onto predefined surfaces and reconstructs the complete DLO routes in the scene by fusing DLO paths across visual fields. Additionally, our method addresses the challenges of DLO crossings and bifurcations.

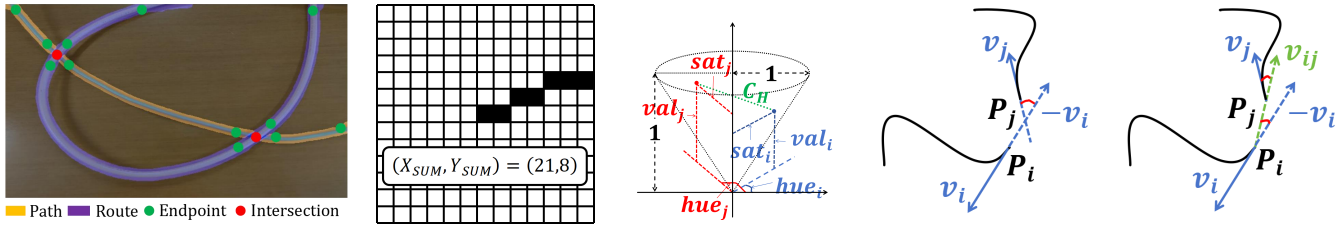
III. Method

Our cross-visual-field BDLO route estimation method consists of three main stages. First, DLO paths are extracted from individual images, and the connection relationships among them are established. Next, the DLO paths are projected onto spatial surfaces using camera pose data and are fused across visual fields. Finally, a route search algorithm based on restricted undirected graphs is applied to recover complete BDLO routes.

A. DLO Path Extraction and Endpoint Pairing

In this section, we propose a strategy to handle crossings and bifurcations in DLO instance segmentation. Fig. 2(a) illustrates the outputs of each step in our method, with DLO paths highlighted for clarity. The pipeline comprises the following key steps. First, we follow prior works to extract DLO masks, skeletons, and key points [17], [18], [19]. Next, we combine the color image with the refined skeleton map to design an effective endpoint pairing cost function. Additionally, we also propose a unified strategy for handling intersections with varying numbers of endpoints, which distinguishes our approach from prior work.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) (b) (c) (d) (e)

Fig. 2. (a) Interpretation of some terms; (b) Endpoint window; (c) HSV cost; (d) Direction cost; (e) Curvature cost.

We employ *Deeplabv3+* for DLO semantic segmentation, which has demonstrated promising results in related studies [16], [17], [18]. Following FASTDLO, we apply the thinning algorithm from [31] to extract the skeleton map M_{sk} from M_{dlo}, ensuring the skeleton is one-pixel wide. Then, a skeleton pixel classification kernel,

$$K = [[1,1,1],[1,10,1],[1,1,1]]^{\top} \tag{1}$$

is convolved with M_{sk}; i.e., $M_{sk} \otimes K$ to classify pixel types. Pixels with a value of 11 (indicating one neighbor) are classified as endpoints, those greater than 12 (three or more neighbors) as intersections, and those with a value of 12 as transition points. Nearby intersections are merged by replacing them with their center points, using a merge threshold defined as $KS = max(W,H)//200$, where W and H are the input image dimensions. For any intersection, its distance to the nearest zero-valued pixel is computed and used as the radius to draw a zero-valued circular region, splitting M_{sk} into several disjoint curves.

Next, we adopt the *walks* algorithm from *Ariadne* to identify connected skeleton pixels and form DLO paths, each of which terminates at two endpoints. For each intersection, the associated endpoints are identified based on their connected paths. Endpoints not associated with any intersection are added to a collection of isolated endpoints.

Accurate endpoint connection is critical for distinguishing DLO routes. The location of $Endpoint_i$ is the essential feature, denoted by $F_{loc,i}$. Next, we design a direction feature $F_{dir,i} = v_i = (X_{sum}, Y_{sum})$ to capture the orientation of $Endpoint_i$. As shown in Fig. 2(b), the black grids in the window centered on the endpoint show the corresponding path. X_{sum} and Y_{sum} are the sums of horizontal and vertical coordinates of all black grids. The *HSV* value of the endpoint pixel from the source image is used as the color feature $F_{color,i}$.

In this paper, we design an endpoint pairing cost function $C_P(\cdot)$ consisting of three independent cost functions to measure the feasibility of connecting any two endpoints.

(1) *HSV* Cost C_H: Measure the differences in the color feature between two endpoints. As shown in Fig. 2(c), the color feature $F_{color,i} = (hue_i, sat_i, val_i)$ is represented in an inverted conic coordinate system of height and radius 1 as $P_{color,i}$. The normalized L2 distance between $(P_{color,i}, P_{color,j})$ is considered as the *HSV* Cost.

$$P_{color,i} = (val_i, sat_i \cdot \cos(hue_i), sat_i \cdot \sin(hue_i)) \tag{2}$$

$$P_{color,j} = (val_j, sat_j \cdot \cos(hue_j), sat_j \cdot \sin(hue_j)) \tag{3}$$

$$C_H(P_i, P_j) = \frac{\|P_{color,i} - P_{color,j}\|_2}{2} \tag{4}$$

(2) Direction Cost C_D: Measure and normalize the error between the direction vectors (v_i, v_j) as shown in Fig. 2(d).

$$C_D(P_i, P_j) = \frac{1 - \frac{-v_i \cdot v_j}{\|v_i\|_2 \cdot \|v_j\|_2}}{2} \tag{5}$$

(3) Curvature Cost C_C: Measure and normalize the minor error between the direction vectors (v_i, v_j) and the vector v_{ij} from P_i to P_j, as shown in Fig. 2(e).

$$C_C(P_i, P_j) = \frac{1 - \max\left(\frac{-v_i \cdot v_{ij}}{\|v_i\|_2 \cdot \|v_{ij}\|_2}, \frac{v_j \cdot v_{ij}}{\|v_j\|_2 \cdot \|v_{ij}\|_2}\right)}{2} \tag{6}$$

The pairing cost function (Pairing Cost C_P) combines the above cost functions:

$$C_P = C_H + C_D + C_C \tag{7}$$

For each intersection, we identify all connectable endpoint pairs, referred to as "*End Pairs*". Fig. 3 shows intersection results with varying numbers of endpoints, including bifurcations and crossings. We first number all endpoints at the intersection and compute pairing costs for all combinations. The combination with the smallest cost C_P is selected, added to *End Pairs*, and removed from the search domain. This process repeats until all endpoints are included in *End Pairs*. As shown in Fig. 3, yellow-highlighted endpoint pairs represent the correct connections selected by our method.

For overlapping connections, we define $C_{H,conn}$ using the endpoints (P_1, P_2) and the midpoint P_{1_2} of the connection. The connection with the smallest $C_{H,conn}$ is prioritized.

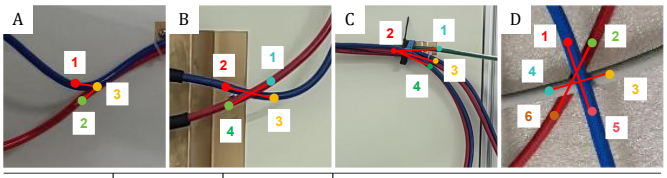
$$C_{H,conn} = C_H(P_1, P_{1\_2}) + C_H(P_{1\_2}, P_2) \tag{8}$$

The intersection layout is used to validate the performance of *CVF-DLO* on the unbranched DLO dataset.

B. DLO Path Projection and Cross-View Fusion

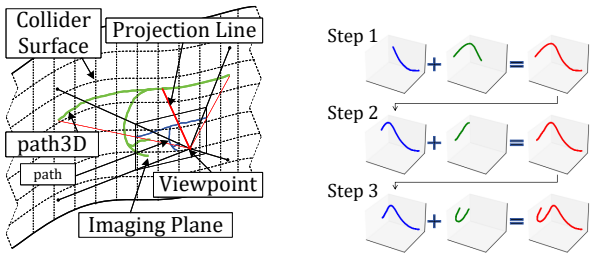
Building on the extracted DLO paths, we propose a method to estimate DLO paths across visual fields. Our method targets cables laid along physical surfaces such as walls, desktops, floors, and ceilings. We define the virtual counterparts of these surfaces as cable-laying surfaces and assume cables lie along these surfaces.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



A		B		C		D					
pair	C_P	pair	C_P	pair	C_P	pair	C_P	pair	C_P	pair	C_P
12	1.31	12	0.63	12	0.30	12	1.76	24	0.63	45	1.45
13	0.27	13	1.67	13	1.56	13	1.34	25	0.64	46	1.53
23	0.48	14	0.10	14	2.08	14	1.10	26	0.13	56	2.05
		23	0.04	23	0.69	15	0.07	34	0.01		
		24	1.81	24	0.14	16	0.66	35	0.97		
		34	0.43	34	2.30	23	1.81	36	0.73		

Fig. 3. Results of intersections with different numbers of endpoints.



(a) (b)

Fig. 4. (a) DLO path3D estimation based on projection; (b) Process of merging DLO path3Ds.

By mapping DLO paths from images onto the cable-laying surfaces, we obtain their spatial coordinates relative to the camera. Next, projection lines are computed from the camera's optical center through the DLO points, based on camera pose. Finally, treating the cable-laying surfaces as collider surfaces, we compute the intersections of the projection lines with them. These intersection points form the scatter representations of spatial DLO paths (see Fig. 4(a)), referred to as path3Ds.

Depth discontinuities and abrupt surface turns may cause misalignment among otherwise continuous DLO projection points, hindering subsequent route estimation. To address this, we fit the path3Ds with quasi-uniform cubic 3D B-spline curves. The parametric form for the B-spline curve is given by:

$$C(u) = \sum_{i=0}^{m} P_i N_{i,3}(u) \tag{9}$$

P_i are 3D control points. There are $m+1$ control points, where $m = max(round(N_{path/100}), 6)$. $N_{i,3}$ is the cubic B-spline basis function. u is the parameter in $[u_0, u_m]$.

Next, we merge disconnected path3Ds caused by field-of-view limitations. We first assign global labels to path3Ds and their endpoints. For each image, we search for the nearest perspectives, ensuring the angle between the adjacent and target perspectives is smaller than the horizontal field of view (FOV) of the camera.

Path3D matching and merging are performed only between two intersecting visual fields. For a path3D pair $(path3D_1, path3D_2)$, we obtain their parametric equations $C_1(u)$ and $C_2(v)$. Then, we solve the following equation:

$$\|C_1(u) - C_2(v)\|_2 < \epsilon \tag{10}$$

To account for the effects of cable suspension and camera pose deviation on projection results, a tolerance is introduced. In our experiments, ϵ is set to 15 mm. The equation is solved using numerical methods. $C_1(u)$ and $C_2(v)$ are discretized into sequential point sets S_1 and S_2. A KD-Tree is constructed from S_1, and for each point in S_1, its nearest neighbor in S_2 is identified. Point pairs with distances less than ϵ are considered adjacent. If the adjacent point pairs are continuous and span a sufficient length along the curve, the two path3Ds are treated as part of the same DLO instance.

Fig. 4(b)illustrates how multiple path3Ds are merged into a single path3D. The merged path3D (red curve, last column) will continue to be used for fusion until all connected path3Ds are combined. During merging, the corresponding endpoint information is also integrated. The connections in *End Pairs* are extended to apply across merged path3Ds.

C. DLO Route Search

The previous steps yield all DLO path3Ds and associated *End Pairs*. To reconstruct the complete route of a DLO instance across views, path3Ds must be connected in the correct order. Our research targets harnesses and bifurcations, allowing a single path to belong to multiple possible routes. We thus adopt a depth-first search strategy on restricted undirected graphs to perform DLO route search.

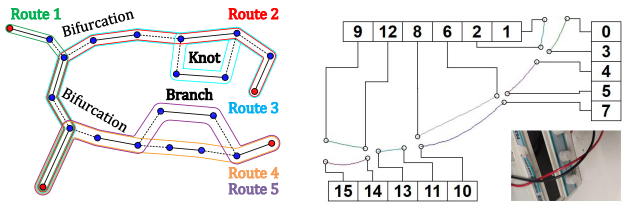
To aid understanding, we first analyze the problem at the image level and apply our algorithm to DLO instance segmentation. A schematic cable relationship diagram is shown in Fig.5(a), where red dots denote isolated endpoints (starting points of DLO routes), and blue dots denote non-isolated endpoints, typically located at intersections. Solid lines indicate DLO paths; dashed lines represent connections between *End Pairs*. Algorithm 1 describes the DLO route search procedure, which starts from an isolated endpoint and alternates between solid and dashed lines until another isolated endpoint is reached. Paths may be traversed repeatedly within a single route, but each connection is used at most once. Bifurcations, branches, and knots will increase the number of possible routes. As illustrated in Fig. 5(a), five potential DLO routes are shown, each marked with a colored bounding box. The algorithm is readily extended to search the complete DLO route network across multiple views.

We further propose a route optimization scheme based on color consistency for unbranched DLO images. The HSV cost $C_{H,ends}$ is computed between the starting and ending points of a given route. Then, the average HSV cost $C_{H,avg}$ between the midpoints of the first path and the other paths in the route is computed. The route consistency cost C_R is:

$$C_R = 100 \times C_{H,ends} + C_{H,avg} \tag{11}$$

By comparing $C_{H,ends}$, routes with inconsistent colors at both ends are filtered out. Accordingly, a large weight is assigned to $C_{H,ends}$ in Eq. 11. $C_{H,avg}$ is then compared to

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) (b)

Fig. 5. (a) Schematic diagram of DLO route searching algorithm; (b) Extracted paths and labeled endpoints of two parallel crossing cables.

TABLE I
ROUTE CONSISTENCY COSTS OF THE POSSIBLE DLO ROUTES

Route	$C_{H,ends}$	$C_{H,avg}$	C_R
$[0, 3, 4, 5, 6, 8, 11, 13, 12, 9]$	0.443	0.505	44.814
$[0, 3, 4, 5, 6, 8, 11, 13, 14, 15]$	**0.044**	**0.405**	**4.767**
$[0, 3, 4, 5, 7, 10, 11, 13, 12, 9]$	0.443	0.522	44.831
$[0, 3, 4, 5, 7, 10, 11, 13, 14, 15]$	0.044	0.422	4.784
$[1, 2, 4, 5, 6, 8, 11, 13, 12, 9]$	0.130	0.190	13.169
$[1, 2, 4, 5, 6, 8, 11, 13, 14, 15]$	0.557	0.332	55.988
$[1, 2, 4, 5, 7, 10, 11, 13, 12, 9]$	**0.130**	**0.096**	**13.075**
$[1, 2, 4, 5, 7, 10, 11, 13, 14, 15]$	0.557	0.228	55.894

determine the most consistent route for each starting endpoint. Fig. 5(b) illustrates the problem: two cables cross twice, resulting in long parallel segments that cannot be pruned at the intersection. Using the above route search algorithm, eight possible routes (expressed as endpoint combinations in Table I) can be identified. If two routes yield the same C_R, the longer one is selected.

Algorithm 1: DLO Route Search Pseudocode

```
Data: EndsDict
Data: Routes
routes = [];
for i in EndsDict.keys() do
    if EndsDict[i]['Type'] ==' iso' then
        traversed = [];
        def MergeRoutes(base, add):
            merge = [];
            for b in base do
                for a in add do
                    merge.append(b + a);
            return merge;
        def GetPathEnd(curr):
            next = EndsDict[curr]['PathEnd'];
            if EndsDict[next]['Type'] ==' iso' then
                return [[next]];
            else
                return MergeRoutes([[next]],
                GetPairEnd(next));
        def GetPairEnd(curr):
            nexts = EndsDict[curr]['PairEnds'];
            re_base = [];
            for next in nexts do
                end_pair = sorted(curr, next);
                if end_pair not in traversed then
                    traversed.append(end_pair);
                    re_base_list+ =MergeRoutes(
                    [[next]],GetPathEnd(next));
                    traversed.remove(end_pair);
            return re_base;
        routes+ =MergeRoutes([[i]],
        GetPathEnd(i));
return routes;
```

IV. EXPERIMENTAL VALIDATION

A. Dataset

We use two public unbranched DLO datasets: *EWD* (Electric Wires Dataset from *Ariadne+*) and *SBHC* (Simple Background, High Curvature Dataset from *mBEST*). *EWD* is a complex background, simple DLO dataset, divided into C1, C2, and C3, with increasing background complexity. *SBHC* is a simple background, highly curved DLO dataset, divided into S1, S2, and S3, where the numbers correspond to the number of cables. To evaluate our method on branched DLOs, we provide a new BDLO dataset, *BWH* (Branched Wiring Harnesses Dataset), which contains 75 images of size 896×504. *BWH* features simple backgrounds and wiring harnesses with frequent bifurcations and crossings. It is divided into B2, B3, and B4, where the tier number indicates the maximum number of cables in each image. Each image includes a binary mask for all cables and separate route masks for individual cables.

In addition to the three planar DLO datasets, we provide a cross-visual-field BDLO dataset. We constructed a nearly circular platform with a 2.5 m diameter modeled after a rocket cabin, and deployed six wiring harnesses (14 cables) along the bulkhead. We uniformly adopted 6.6 mm diameter copper conductor power cables, ranging in length from 2 to 7 m. The designed routes included parallel paths and bifurcations. Both virtual and physical platforms were provided, with the physical cables laid according to the virtual design. However, discrepancies exist between the real and designed routes due to the cable's physical characteristics.

We used Hololens2 to capture images at a resolution of 896×504. To ensure complete coverage of the cables, 20 images were collected, considering scene size and image resolution. Frontal perspectives were adopted, with overlapping areas between adjacent fields of view. Hololens2 provided camera poses relative to the platform through its built-in IMU and scene features tracking algorithm. We used the same camera poses and parameters in the virtual scene to capture images of virtual cables. Finally, we constructed the *LABD* dataset, which includes 20 virtual cable images (*LABD-Syn*) and 20 real cable images (*LABD-Real*), along with camera poses, platform poses, and cable models.

B. Training on DLO semantic segmentation

We tested *Deeplabv3+* with different backbones for semantic segmentation. *EWD* provided 28593 synthetic samples for training and 135 samples for testing. *Deeplabv3+* achieved an IoU of 0.723 with *Resnet50* and 0.822 with *Resnet101* on

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

EWD. The model was also applied to BWH, achieving an IoU of 0.598 with *Resnet50* and 0.807 with *Resnet101*. We collected 50 photos of the cabin platform and labeled the cable masks. We synthesized 10000 samples from 30 *LABD-Real* photos for fine-tuning, with the remaining 20 for testing. The batch size was 16, the learning rate was 0.01, and the crop size was 513. After 30000 iterations, the fine-tuned model (*Deeplabv3+* with *Resnet101*) achieved an IoU of 0.879 on *LABD-Real*. Training and evaluation were conducted on a desktop with a 2.90 GHz Intel(R) Core(TM) i7-10700 CPU and 16GB RAM.

C. Evaluation of DLO Instance Segmentation

We combined the methods from III-A and III-C to implement DLO instance segmentation in images and compared it with several optimal methods, including *FASTDLO*, *RT-DLO*, and *mBEST*. First, we evaluate the endpoint pairing cost function from III-A for handling intersections on *EWD* and *SBHC*. As shown in the first six rows of Fig. 6, our method effectively handles DLO intersections, including cases such as multiple DLO crossings (C2), small-angle crossings (C3, S3), and knots (S1, S3), demonstrating the universality and robustness of our approach.

Additionally, we conducted experiments on *BWH* to validate the DLO route search algorithm from III-C. The same DLO path may belong to multiple DLO routes in the BDLO images. Therefore, we used multiple independent DLO route masks, instead of a single colored mask, to represent the BDLO routes. We modified the above methods to plot the predicted DLO routes separately. As shown in the last three rows of Fig. 6, only our method correctly handles bifurcations and extracts the BDLO routes. Notably, we applied the route optimization scheme from III-C only to unbranched DLO images. For BDLO images, our method enumerates all potential routes, including those beyond the existing ones (B3). Screening these routes requires more information beyond the cable's characteristics.

We used two metrics to evaluate the accuracy of DLO instance segmentation. The Dice Similarity Coefficient (*DICE*) measures the consistency between the predicted DLO mask and the binary mask of M_{gt}. However, *DICE* only reflects the semantic segmentation errors. Inspired by the PQ loss, we designed PQ_{dlo} to represent the accuracy of DLO route segmentation, applicable to both DLO and BDLO samples. Assuming there are M DLO instances in the predicted image, $pred(p)$ represents all the predicted DLO pixels, and $pred(p)_{i,1\leq i\leq M}$ represents the pixels of the i-th DLO instance. There are N DLO instances in M_{gt}. $target(t)$ represents all the DLO pixels, and $target(t)_{j,1\leq j\leq N}$ represents the pixels of the j-th DLO instance.

$$DICE = \frac{2 \times |p \cap t|}{|p| + |t|} \tag{12}$$

$$PQ_{dlo} = \frac{1}{|p \cup t|} \sum_{j} \max_{i} \left(|p_i \cap t_j| \cdot \frac{|p_i \cap t_j|}{|p_i \cup t_j|} \right) \tag{13}$$

The average *DICE* and PQ_{dlo} of four methods with different masks on *SBHC*, *EWD*, and *BWH* are shown in Table II.

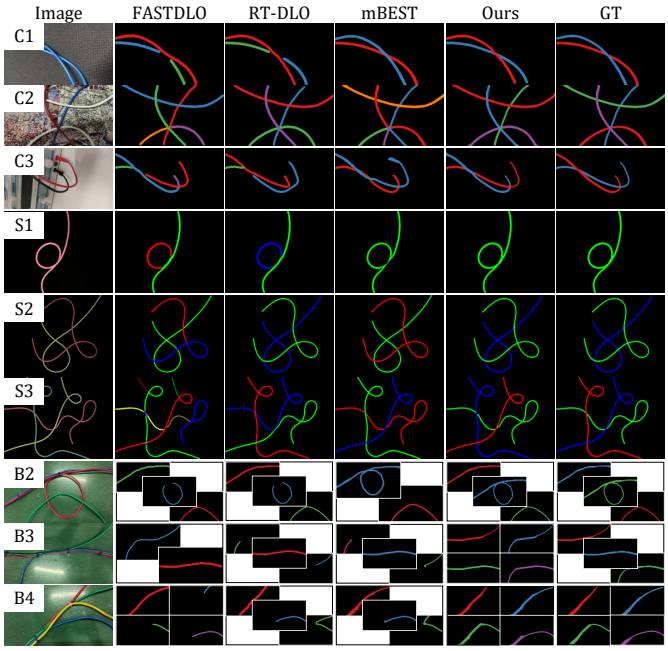


Fig. 6. Performance of different pipelines on DLO instance segmentation.

CVF-DLO achieves state-of-the-art performance with high-precision masks. In cases of mask degradation, *RT-DLO*, using discrete points, performs best, while *CVF-DLO* still outperforms other methods using skeleton pixels.

Among methods using skeleton pixels, the primary differences lie in the intersection clustering and matching functions. We propose an endpoint pairing cost function and a route search algorithm for this. *FASTDLO* adopts a similarity network, and *mBEST* uses the minimum bending energy. While considering bifurcations and route search, our method remains concise and efficient. Overall, the running speed of *CVF-DLO* is similar to *mBEST* when handling unbranched DLO images.

Additionally, we conducted ablation experiments on the endpoint pairing cost function by testing different sub-cost components and their combinations through PQ_{dlo}. All experiments in Table III use *Deeplabv3+* with *ResNet101* as the segmentation model. As shown in Table III, C_D and C_C play a dominant role, and removing any sub-cost results in a performance drop. On *SBHC*, using only C_D or C_C is sufficient to achieve the best performance.

D. Evaluation of DLO Route Estimation

We conducted experiments on *LABD* and performed cross-visual-field DLO route estimation using *LABD-Syn* and *LABD-Real*, respectively. First, we extracted the cable paths and connections from the images in *LABD*. Then, by projecting the cable paths onto collider surfaces, concatenating them across visual fields, and applying the DLO route search algorithm to the path3Ds across the entire scene, we enumerated all possible cable routes. *LABD* contains six wiring harnesses (W01-06) and a total of 14 cables. After estimating the cable routes using our pipeline, 84 possible routes were obtained, 14 of which correctly matched the actual cables. This validates

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II
PERFORMANCE OF DIFFERENT PIPELINES ON DLO SEGMENTATION

Pipeline	Mask	Backbone	Skeleton	*DICE*(%)	PQ_{dlo}(%)	FPS
		SBHC				
FASTDLO	Color	–	✓	96.19	63.38	11.25
RT-DLO	Color	–	×	98.56	92.63	11.19
mBEST	Color	–	✓	99.39	97.20	18.96
CVF-DLO	Color	–	✓	99.67	96.69	19.18
		EWD				
FASTDLO	DLv3+	R50	✓	79.62	36.10	11.01
FASTDLO	DLv3+	R101	✓	85.88	55.17	10.88
FASTDLO	GT	–	✓	89.56	55.32	24.06
RT-DLO	DLv3+	R50	×	81.28	44.17	18.21
RT-DLO	DLv3+	R101	×	87.90	59.35	14.37
RT-DLO	GT	–	×	93.49	73.65	30.32
mBEST	DLv3+	R50	✓	75.42	29.31	17.65
mBEST	DLv3+	R101	✓	88.73	60.92	15.58
mBEST	GT	–	✓	95.21	71.00	58.76
CVF-DLO	DLv3+	R50	✓	77.69	44.44	17.93
CVF-DLO	DLv3+	R101	✓	89.25	65.31	15.31
CVF-DLO	GT	–	✓	95.09	80.45	57.73
		BWH				
FASTDLO	DLv3+	R50	✓	70.07	12.77	7.58
FASTDLO	DLv3+	R101	✓	85.18	31.88	6.82
FASTDLO	GT	–	✓	88.77	35.57	11.84
RT-DLO	DLv3+	R50	×	69.57	17.38	11.31
RT-DLO	DLv3+	R101	×	85.74	32.21	12.84
RT-DLO	GT	–	×	90.70	37.46	24.15
mBEST	DLv3+	R50	✓	66.67	12.81	19.10
mBEST	DLv3+	R101	✓	84.42	23.04	15.16
mBEST	GT	–	✓	91.06	21.29	27.27
CVF-DLO	DLv3+	R50	✓	68.98	16.72	10.18
CVF-DLO	DLv3+	R101	✓	87.56	59.40	9.13
CVF-DLO	GT	–	✓	94.49	71.87	25.18

DLv3+: *Deeplabv3+*; R50: *Resnet50*; R101: *Resnet101*

TABLE III
ABLATION STUDY OF ENDPOINT PARING FUNCTION

	C_D	C_C	C_H	C_DC_C	C_DC_H	C_CC_H	$C_DC_CC_H$
EWD	64.99	64.32	54.30	64.25	64.69	64.87	65.31
BWH	57.38	56.89	42.15	57.40	52.93	52.91	59.40

the DLO route search strategy from III-C in scene-level cable route reconstruction. The correct cable routes can be further filtered by considering the cable's color and number. The estimated cable routes based on *LABD-Real* are shown in Fig. 7.

We quantitatively analyzed the error distribution between the estimated routes based on *LABD-Syn* and the designed routes. The cable route reconstruction error primarily stems from III-B. We uniformly sampled 200 points along the estimated route and identified their nearest points on the designed route, using the distances as the estimation error. As shown in Fig. 8, the estimated cable routes align with the designed routes at the scene scale. In the middle section of the cable, where it is roughly laid along the bulkhead,

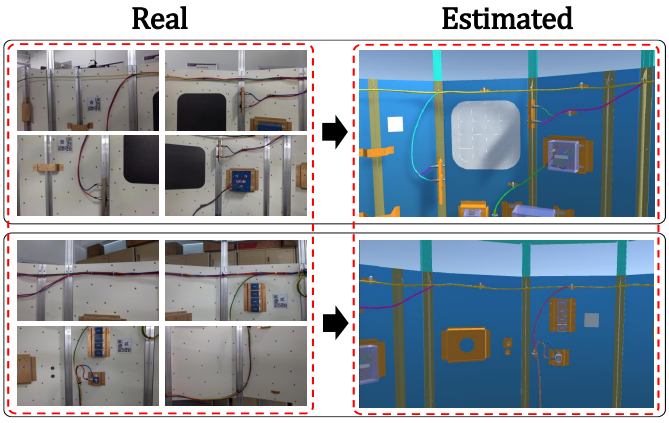


Fig. 7. Cross-visual-field route estimation of BDLOs in *LABD-Real*.

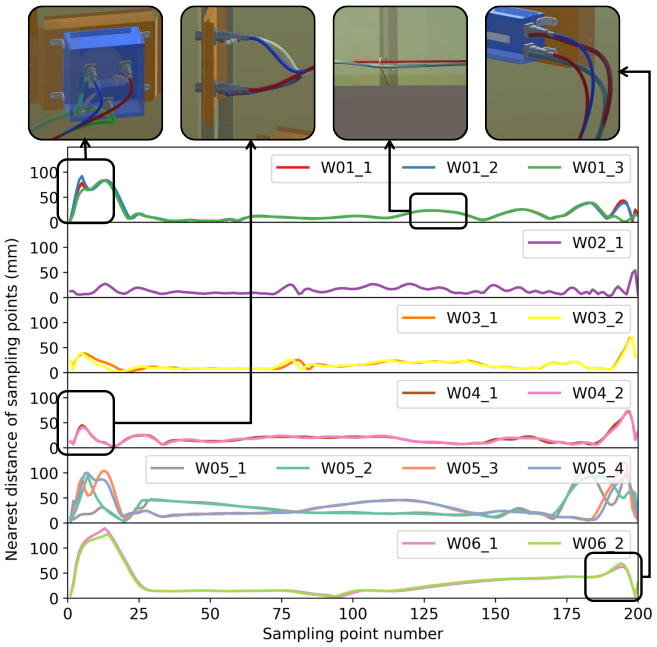


Fig. 8. Error distribution of Cable Route Estimation based on *LABD-Syn*.

the estimation error is within 20mm for most samples. However, the suspended portions, especially those connected to the joints, exhibit relatively large estimation errors. Despite this, the connection point between the cable and the joint is accurately located.

The errors in cable route estimation primarily result from cable suspension, deviations in camera attitude, and the fitting of the estimated route with the B-spline. While the projection-based method using mobile devices cannot match the accuracy of high-precision cameras mounted on robotic arms or LiDAR systems, it offers advantages in flexibility and computational efficiency. We conducted experiments on the cross-view path fusion threshold and camera pose perturbations. The results and corresponding analysis are provided in the supplementary video.

V. Conclusions

This paper introduces *CVF-DLO*, a method for estimating the complete routes of branched deformable linear objects laid along surfaces from multiple overlapping views. First, we propose an endpoint pairing cost function and an intersection processing strategy to extract DLO paths and connections from images. Second, using camera poses and scene information, we project the DLO paths onto preset surfaces and fuse those broken paths due to field-of-view limitations. Third, we present a DLO route search strategy for harnesses that handles bifurcations, branches, and knots, and enumerates all possible cables. By combining path extraction and route search, we construct a pipeline for DLO instance segmentation, improving upon state-of-the-art in both accuracy and robustness, particularly for branched DLOs. Finally, we apply the route search algorithm to all DLO paths in the scene to achieve cross-visual-field route estimation. We provide a planar BDLO dataset and a cross-visual-field wiring harness dataset. Experiments show that *CVF-DLO* provides a cost-effective BDLO route estimation method capable of handling complex BDLOs and reconstructing the B-spline model of each potential DLO instance.

In its current form, *CVF-DLO* is better suited for DLOs laid along physical surfaces. For estimating suspended cables, incorporating joint directions and the cable's physical characteristics into the fitting process can significantly improve accuracy. Additionally, *CVF-DLO* currently provides only potential DLO route choices. To identify the correct and unique route, further information, such as color and numerical labeling, must be incorporated to filter the options.

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, *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, (2010) 490–495.

[2] H.G. Nguyen, M. Kuhn, J. Franke, Manufacturing automation for automotive wiring harnesses, *Procedia CIRP*, 97 (2021) 379-384.

[3] 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. *The International Journal of Robotics Research*. 37(7) (2018) 688-716.

[4] J. Zhu, B. Navarro, P. Fraisse, A. Crosnier and A. Cherubini, Dual-arm robotic manipulation of flexible cables, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, (2018) 479-484.

[5] M. Yan, Y. Zhu, N. Jin and J. Bohg, Self-Supervised Learning of State Estimation for Manipulating Deformable Linear Objects, *IEEE Robotics and Automation Letters*, 5(2) (2020) 2372-2379.

[6] A. Keipour, M. Bandari, and S. Schaal, Deformable one-dimensional object detection for routing and manipulation, *IEEE Robotics and Automation Letters*, 7(2) (2022) 4329–4336.

[7] D. Tong, A. Borum and M. K. Jawed, Automated Stability Testing of Elastic Rods With Helical Centerlines Using a Robotic System, *IEEE Robotics and Automation Letters*, 7 (2022),

[8] D. D. Gregorio, G. Palli, and L. D. Stefano, Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation, 2018 Asian Conference on Computer Vision, (2018).

[9] H.G. Nguyen, J. Franke, Deep learning-based optical inspection of rigid and deformable linear objects in wiring harnesses, *Procedia CIRP*, 104 (2021), 1765-1770.

[10] M. Denninger et al., Blenderproc, *CoRR*, abs/1911.01911 (2019).

[11] W. Qiu and A. Yuille, UnrealCV: Connecting computer vision to unreal engine, *Proceedings of the European conference on computer vision (ECCV)*, (2016) 909–916.

[12] 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 Robotics and Automation Letters*, 8(2) (2023) 1013-1020.

[13] J. Dirr, D. Gebauer, J.J. Yao and R. Daub, Automatic Image Generation Pipeline for Instance Segmentation of Deformable Linear Objects. Sensors, 23(6) (2023) 3013.

[14] R. Zanella, A. Caporali, K. Tadaka, D.D. Gregorio and G. Palli, Auto-generated Wires Dataset for Semantic Segmentation with Domain-Independence, *2021 International Conference on Computer, Control and Robotics (ICCCR)*, Shanghai, China, (2021) 292-298.

[15] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, Encoderdecoder with atrous separable convolution for semantic image segmentation, *Proceedings of the European conference on computer vision (ECCV)*, (2018) 801–818.

[16] A. Caporali, R. Zanella, D.D. Greogrio and G. Palli, Ariadne+: Deep Learning–Based Augmented Framework for the Instance Segmentation of Wires, *IEEE Transactions on Industrial Informatics*, 18(12) (2022) 8607-8617.

[17] A. Caporali, K. Galassi, R. Zanella, and G. Palli, FASTDLO: Fast deformable linear objects instance segmentation, *IEEE Robotics and Automation Letters*, 7(4) (2022) 9075–9082.

[18] A. Caporali, K. Galassi, B. L. Zagar, R. Zanella, G. Palli, and A.C. Knoll, "Rt-dlo: Real-time deformable linear objects instance segmentation, *IEEE Transactions on Industrial Informatics*, 19(11) (2023) 11333-11342.

[19] A. Choi, D. Tong, B. Park, D. Terzopoulos, J. Joo, and M.K. Jawed, mBEST: Realtime Deformable Linear Object Detection Through Minimal Bending Energy Skeleton Pixel Traversals, *IEEE Robotics and Automation Letters*, 8(8) (2023) 4863-4870.

[20] P. Kicki, M. Bednarek, P. Lembicz, G. Mierzwiak, A. Szymko, M. Kraft, and K. Walas, Tell Me, What Do You See?—Interpretable Classification of Wiring Harness Branches with Deep Neural Networks. *Sensors*, 21(13) (2021) 4327.

[21] M. Zürn et al., Deep Learning-Based Instance Segmentation for Feature Extraction of Branched Deformable Linear Objects for Robotic Manipulation, *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, Auckland, New Zealand, (2023) 1-6.

[22] M. Zürn, M. Wnuk, A. Lechler and A. Verl, Topology Matching of Branched Deformable Linear Objects, *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, (2023) 7097-7103.

[23] I. Mishani, A. Sintov, Learning configurations of wires for real-time shape estimation and manipulation planning, *Engineering Applications of Artificial Intelligence*,121 (2023), 105967.

[24] K. Lv, M. Yu, Y. Pu, X. Jiang, G. Huang and X. Li, Learning to Estimate 3-D States of Deformable Linear Objects from Single-Frame Occluded Point Clouds, *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, (2023) 7119-7125.

[25] Y. Zhu, X. Xiao, Wu, W. *et al.* 3D Reconstruction of deformable linear objects based on cylindrical fitting, *Signal, Image and Video Processing*, 17 (2023) 2617–2625.

[26] Z.M. Fan, L. Guo, Y.B. Chen, F. Xiao and J.H. Zhang, CPC: Cable Parameters Calculation Based on 3D Point Cloud, Intelligent Robotics and Applications, (2022) 13458.

[27] D. Chiaravalli, A. Caporali, A. Friz, R. Meattini and G. Palli, A Vision-based Shared Autonomy Framework for Deformable Linear Objects Manipulation, *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Seattle, WA, USA, (2023) 733-738.

[28] W. Wu, Y. M. Zhu, X. J. Zheng, Y. Guo, A novel cable-grasping planner for manipulator based on the operation surface, *Robotics and Computer-Integrated Manufacturing*, 73 (2022) 102252.

[29] A. Caporali, K. Galassi and G. Palli, Deformable Linear Objects 3D Shape Estimation and Tracking From Multiple 2D Views, *IEEE Robotics and Automation Letters*, 8(6) (2023) 3852-3859.

[30] K. Galassi, A. Caporali and G. Palli, Cable Detection and Manipulation for DLO-in-Hole Assembly Tasks, *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*, Coventry, United Kingdom, (2022) 1-6.

[31] T. Y. Zhang and C. Y. Suen, A fast parallel algorithm for thinning digital patterns, *Communications of the ACM*, 27(3) (1984), 236–239.