A Robust Deformable Linear Object Perception Pipeline in 3D: From Segmentation to Reconstruction

Sun Zhaole* Hang Zhou Li Nanbo Longfei Chen Jihong Zhu
Robert B. Fisher

November 28, 2023

Abstract

3D perception of deformable linear objects (DLOs) is crucial for DLO manipulation. However, perceiving DLOs in 3D from a single RGBD image is challenging. Previous DLO perception methods fail to extract a decent 3D DLO model due to different textures, occlusions, sparse and false depth information. To address these problems and provide a more robust DLO perception initialization for downstream tasks like tracking and manipulation in complex scenarios, this paper proposes a 3D DLO perception pipeline to first segment a DLO in 2D images and post-process masks to eliminate false positive segmentation, reconstruct the DLO in 3D space to predict the occluded part of the DLO, and physically smooth the reconstructed DLO. By testing on a synthetic DLO dataset and further validating on a real-world dataset with seven different DLOs, we demonstrate that the proposed method is an effective and robust 3D perception pipeline solution with better performance on 2D DLO segmentation and 3D DLO reconstruction compared to State-of-the-Art algorithms.

1 Introduction

Deformable linear objects (DLOs), e.g. ropes, cables, pipes, hoses, and tubes, are found widely in surgical theaters, offices, textile factories, and other industries [1]. A critical application for robots is manipulating these DLOs to perform different tasks like shape control, cable plug-in, and knotting [2]. To better manipulate DLOs, robots need an accurate 3D understanding of the DLO shapes [1, 3]. Though many papers have proposed methods to track DLOs using a sequence of video frames during manipulation [4, 5], they often assumed a clean DLO configuration to start with like a simple curved shape without occlusions, where the DLOs' depth maps are also nearly complete in the first frame. However, these assumptions do not always hold in practice. For instance, Figure 1 shows our pipeline, which takes a common lab configuration of a grasped DLO in the air with occlusions as input like previous works [6, 7, 8]. To accurately obtain a 3D model of a DLO with occlusions, we first obtain 2D DLO masks and then reconstruct the DLOs in 3D. The proposed 3D DLO perception pipeline tackles both the challenges of 2D DLO segmentation and 3D reconstruction in complex scenarios where previous methods only focused on one aspect of the problem and did not solve it as well as proposed here.

To segment a DLO, traditional approaches commonly used color-specific DLOs with a contrasting background for color space segmentation. Structured or simplified 2D working spaces, such as a table with uniform color [9] or 3D workspaces with simple backgrounds, are widely adopted by DLO tracking algorithms [5, 4, 6]. As for DLO segmentation in environments with different backgrounds, learning-based neural network segmentation architectures have been investigated [10, 11, 12]. These works show the importance of DLO data collection. However, collecting and labeling hundreds of real-world DLO images or rendering realistic DLOs in software takes much human effort and performs poorly on previously unseen DLOs with quite different textures (see the top row in Figure 2).

*Sun Zhaole, Li Nanbo, Longfei Chen, and Robert B. Fisher are with the School of Informatics, University of Edinburgh, UK.
Hang Zhou is with School of Mechatronic Engineering and Automation, Shanghai University, China.
Jihong Zhu is with School of Physics, Engineering and Technology, University of York, UK.
Corresponding author: zhaole.sun@ed.ac.uk

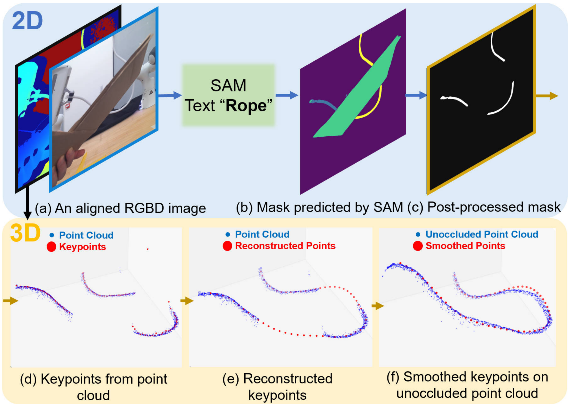
2D
SAM
Text “Rope”
(a) An aligned RGBD image
(b) Mask predicted by SAM
(c) Post-processed mask
3D
Point Cloud
Keypoints
Point Cloud
Reconstructed Points
Unoccluded Point Cloud
Smoothed Points
(d) Keypoints from point cloud
(e) Reconstructed keypoints
(f) Smoothed keypoints on unoccluded point cloud

Figure 1: Our 3D DLO perception pipeline.

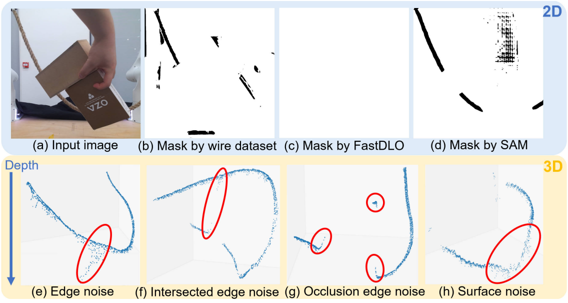

Figure 2: Several DLO perception challenges. **Top:** Low-quality DLO segmentation mask from current approaches. (a) An RGB image of an occluded jute rope. (b) A segmented mask using a segmentation model trained on the wire dataset [10]. (c) A segmented mask using FastDLO [12] (no DLO is detected in this case). (d) A segmented mask using SAM with the text prompt “rope” [18]. **Bottom:** Noisy points caused by the flying pixel issue in the DLO point cloud. (e) Flying pixels between the edge of the DLO and the table. (f) Flying pixels between two self-intersecting parts near each other. (g) Flying pixels between the DLO and the occlusion. (h) Noisy point cloud of reflective DLO surface.

To estimate the DLO state, a common marker-based practice is to model the DLO as connected keypoints [13, 14, 15], which bypasses DLO segmentation and directly estimates the DLO states by attaching markers to the DLOs. Recent markless research, such as by Keipour et al. [16], Kicki et al. [17], and Lv et al. [6], estimate keypoints in 2D and 3D space using segmentation masks and a point cloud. However, we found that the proposed methods do not generalize well to 3D cases [16] or are not robust against depth noise or large occlusions [6, 17] (see noisy point cloud in the bottom row in Figure 2, which caused the incorrect reconstruction in Figure 8).

So far, all mentioned works focused on doing DLO segmentation or DLO state estimation separately, and there is no general and robust solution for 3D DLO perception with mentioned challenges. Existing research in either DLO segmentation or reconstruction also has several unsolved problems. As stated above, sequentially combining the existing algorithms on each part of DLO perception does not lead to an accurate and robust 3D DLO perception result.

To solve both 2D DLO segmentation and 3D DLO reconstruction, the publicly available Segment-Any-Model (SAM) process [18] was used to get DLO masks, and the masks were post-processed to reduce false-positive segmentation (see Section 3.1), obtaining better segmentation performance compared to previous methods on real-world DLO datasets (see Section 4.2). A geometric completion method was used, based on Bezier curves, to connect the DLO across occlusions in the mask (see Section 3.2), achieving a higher reconstruction success rate and lower Chamfer distance between the reconstructed DLO and unoccluded point cloud in both the synthetic dataset and the real-world dataset (see Section 4.3). We then use a Discrete Elastic Rod model (DER) to physically smooth the reconstructed DLO (see Section 3.3), which produced less bending energy on average (see Section 4.3) while still remaining close to the original point cloud.

The contributions of this paper are:

We propose a robust DLO perception pipeline that:

1) Combines a post-processing method with the large vision model SAM to improve the segmentation performance on DLOs by reducing false positive segmentations in the background and occlusions without any additional knowledge of previously unseen DLOs.

2) Produces 3D DLO reconstruction which is robust against sparse depth maps with occlusions and flying pixels.

3) Models the reconstructed DLO based on the DER model to smooth and make it more physically realistic.

Experiments on a synthetic DLO dataset and a real-world dataset of 7 different DLOs show that the proposed method outperforms two State-of-the-Art algorithms on DLO segmentation [10, 12] and reconstruction [6][1].

2 RELATED WORK

Thin Structure Segmentation: Thin structures and methods to segment them are common, from large scales such as road lane lines in autonomous driving [19], drone power line inspection [20], road networks in satellite image analysis [21], to small scales like tubular structures and blood vessels in medical image analysis [22], to cables, tubes, ropes, wires in household and industrial robotic manipulation tasks [16, 11, 23] which is the focus of this paper.

DLO Segmentation: Learning-based methods are widely adopted for DLO segmentation. Zanella et al. [10] used an auto-augmented wire dataset with different backgrounds, Caporali et al. [12] collected rendered DLO images with labels, and Thananjeyan et al. [11] proposed a UAV labeling method on deformable objects to reduce human labeling effort.

DLO State Estimation: Using markers or not are the two common methods for DLO state estimation. A variety of markers are used, including Vicon markers [13], colorful rings painted on the DLO [14], and QR code markers [15]. Research on markerless DLO state estimation has emerged recently, such as by Keipour et al. [16] who proposed an occlusion-aware DLO connection and merging algorithm in 2D space, Kicki et al. [17] used the same method in 3D, and Lv et al. [6] used a deep learning method to directly estimate DLO states from 3D point cloud data with occlusions.

Discrete Elastic Rods: A Discrete Elastic Rod (DER) [24] is a discrete geometric model of thin flexible rods that can represent stiff stretching, bending, and twisting effectively based on stretching, bending, and twisting energies. The DER was initially designed for simulating rods in computer graphics and later generalized for DLO manipulation [25]. Unlike previous approaches, we used the DER model to enforce DLO physical smoothness in 3D after reconstruction.

3 APPROACH

We present the reconstruction pipeline, from 2D DLO segmentation (Section 3.1), 3D DLO reconstruction (Section 3.2) to DLO smoothing (Section 3.3).

3.1 2D DLO Segmentation

Segment-Anything-Model (SAM) [18] is a large, zero-shot segmentation model trained on more than 11M images with labeled masks. This pre-trained model was used without fine-tuning or adding additional DLO training data. After extracting detected object masks using SAM, Grounding-DINO [26] was used, which inputs text prompts and the masks to find (without human supervision) object masks that satisfy the given prompt (e.g. rope). Grounding-DINO is pre-trained on large detection datasets, such as O365 [27], which contain ropes as well as other common objects.

Though SAM on DLO segmentation with text input performs well in most cases, there are some cases containing large areas of false positives arising from the background or from occlusions (see (d) in Figure 2). To reduce the false positive pixels, an effective post-processing method that did not need additional labeled images was used. Masks predicted by SAM having different categories are all regarded as the same DLO, assuming only one DLO to segment in each image. *Detecting multiple*

[1] See a video: https://youtu.be/nFoU-uAYUmg. Implementation codes and test datasets are at: https://github.com/TheGoblinTechies/DLO-perception-pipeline

Algorithm 1: 3D Reconstruction

```
Input  : A 2D mask and point cloud of a DLO
Output: Sorted keypoints DLOPoints in 3D
// 2D morphological operations
$DLOSkeleton \leftarrow$ ExtractSkeleton ($Mask$) ;
$DLOChains \leftarrow$ EliminateConjunctions ($DLOSkeleton$) ;
for $Chain_i$ in $DLOChains$ do
    $Nodes2D_i \leftarrow$ TracePoints ($Chain_i$) ;
end
$Points3D \leftarrow$ KMeans ($PointCloud$, $Centers \leftarrow Nodes2D$) ;
$Nodes3D \leftarrow$ Median ($Points3D$);
$DLOPoints \leftarrow [Nodes3D_1, ..., Nodes3D_n]$ ;
// 3D Reconstruction
for $i \leftarrow 1$ to $n$, $j \leftarrow i$ to $n$ do
    $Cost_{i,j} \leftarrow$ ComputeCost ($Nodes3D_i, Nodes3D_j$) ;
end
$DLOPoints \leftarrow$ ShortestPath ($DLOPoints$, $Cost$) ;
$DLOPoints \leftarrow$ BezierCurve ($DLOPoints$) ;
$DLOPoints \leftarrow$ BSpline ($DLOPoints$) ;
```

DLOs is not our focus and previous research can already separate multiple DLOs in one image [28] based on the DLO semantic masks.

Regarding each connected area of the mask as a component, post-processing of masks from SAM consists of the following (see Figure 3): 1) Remove connected components whose area is below a threshold (e.g. 0.005% pixels). 2) Remove components whose skeletons have irregular numbers of ends and conjunctions: morphological operations are used to count the number of endpoints and conjunctions of each component. The allowable number of endpoints is between 0 and 4, and the number of conjunctions is between 0 and 5. 3) Remove components of irregular width: Extract skeletons of each DLO component mask and estimate the width of the DLO by dividing the area of the component by the total length of the skeleton. Discard components whose width is outside an allowable range.

3.2 3D Reconstruction

3D reconstruction takes the mask and the DLO point cloud cropped by the mask as input. The details of each step are presented in Algorithm 1 and are illustrated in Figure 4.

Extracting keypoints: This step has four operations, *ExtractSkeleton*, *EliminateConjunctions*, *TracePoints*, and *KMeans*, based on Keipour et al's method [16] for the first three operations. In detail: extract a skeleton from the 2D mask to get *DLOSkeleton* and use a 3×3 kernel to find all endpoints and conjunctions based on morphological methods to get *DLOChains* consisting of several chains after splitting the curve at conjunctions. Edge pixels are eroded to create curves with only one-pixel widths. By selecting a suitable pixel length distance (e.g. 10 pixels), the extracted keypoints can be connected to form a chain. However, these operations only provide keypoints in 2D space. Extending these morphological operations into 3D (based on Keipour's approach in 2D) is difficult when the point cloud is noisy and has many flying pixels. To find the 3D keypoints, *KMeans* is used to cluster the point cloud points. The cluster centers become the keypoints (see (a) and (b) in Figure 4). Only the 3D points whose depth is in the closest 25% are clustered, which reduces the influence of flying pixels on the keypoint depth estimates (See (c) in Figure 4). Any chain having less than 10 points is deleted. The two points at each end of a chain are discarded to eliminate flying pixels caused by occlusions, as shown in Figure 2 (g). Once the whole DLO is reconstructed, the two ends are linearly extended with two additional points.

Initial reconstruction: Keipour's method is used to decide which two chains are to be connected and can be directly adapted to 3D with the same cost function. However, their connection strategy is based on a combination of lines and curves with many rules in 2D. These rules are difficult to adapt

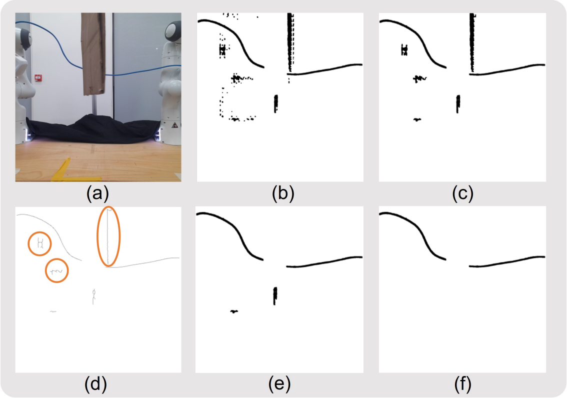

Figure 3: Post-processing the DLO masks from SAM. **(a)** is the image of an occluded Ethernet cable. **(b)** is the mask directly generated by SAM with a text prompt *rope*. The black area is the DLO mask. **(c)** is the mask by removing all components whose area is below a certain threshold. **(d)** is the skeleton extracted from (c), where we circle all irregular skeletons. **(e)** is the mask that removes all components whose skeletons have irregular numbers of ends and conjunctions. **(f)** is the final post-processed mask by removing thick or too thin width components.

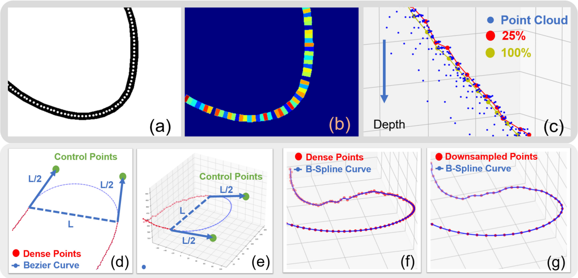

Figure 4: The reconstruction method. **(a)** the mask and the keypoints found using morphological operations. **(b)** the result of K-Means clustering on the 2D mask, where each color represents the mask pixels in the same cluster, and the cluster center is the keypoint in (a). **(c)** a comparison figure where lightgreen points are the median of all points in the same cluster (top 100%), red points are the median of those points having the top 25% smallest depth, and blue points are the original 3D points selected by the segmentation mask. **Bottom: (d)** and **(e)** demonstrate using a Bezier curve to connect the two components of the DLO in 2D and 3D. **(f)** a DLO showing both the dense directly reconstructed 3D keypoints (red), and interpolating B-Spline (blue curve). **(g)** a DLO showing the blue interpolating B-Spline from (f), with sparse downsampled B-Spline points (red).

to 3D space, and may lead to different connections under the same scenario as they showed in their 2D examples. Besides, a combination of lines and curves is not smooth enough with non-continuous derivatives. A similar work, DLOFTBs, used B-Splines to do 3D reconstruction [17] (whereas we use B-Splines for identical distance downsampling - presented next). With a large gap between two chains' ends, directly applying B-Spline connection makes the interpolated curve straight, which leads to a performance drop when the curve should have a large curvature. To avoid this, a concise and simple connection strategy that is easy to follow is proposed. A Bezier curve with two control points is used to connect pieces with a smoother and more deterministic completion (see Figure 4 (d) and (e)).

B-Spline Downsampling: DLO keypoints are not located evenly in 3D when extracted from the 2D masks due to different sample point depths. Further, more than one hundred keypoints are typically reconstructed, whereas previous approaches to DLO state estimation usually yielded much fewer, e.g. 64 [29] or 50 [6] points. Downsampling the keypoints and distributing them approximately equally along the DLO is desirable. Thus, a continuous B-Spline is fitted to the sorted keypoints, which is then downsampled to reduce the number of DLO keypoints to 60 with a more even distribution (see Figure 4 (f) and (g)).

3.3 Discrete Elastic Rod (DER) Fitting.

The previous step extracts the DLO keypoints from the point cloud but does not guarantee the 3D physical smoothness of the DLO, due to the depth sensor noise and flying pixels.

It is hard to define the smoothness of a DLO or how physically realistic the DLO is. We use the Discrete Elastic Rod (DER) computer graphics model [24] where 3D curves and rods with minimal energy are considered to be smooth and physically realistic enough [30]. Fitting a DER trades off the 3D fitting error and a fitting energy penalty. We are not the first to use a DER model, e.g. Lv et al. [25] used a DER model for DLO shape control. Here, the DER model does shape smoothing. A DER-modeled DLO consists of n keypoints with 3D position coordinates and $n-1$ cylindrical segments between keypoints with a sectional radius r. The total energy terms of a DER are split into stretching, bending, and twisting energy, $E = E_s + E_b + E_t$:

$$E_s = \frac{1}{2}\sum_{i=1}^{i=n-1} \frac{\pi Y r^2}{|\bar{s}_i|}(|s_i| - |\bar{s}_i|)^2 \tag{1}$$

$$E_b = \frac{1}{2}\sum_{i=2}^{i=n-1} \frac{\pi Y r^4}{4|\bar{l}_i|}(\kappa_i - \bar{\kappa}_i)^2 \tag{2}$$

$$E_t = \frac{1}{2}\sum_{i=1}^{i=n-1} \frac{\pi Y r^4}{4(1+\nu)|\bar{l}_i|}(\lambda_i - \bar{\lambda}_i)^2, \tag{3}$$

where Y is Young's modulus, ν is the Poisson ratio, usually set to $\nu = 0.5$. $|s_i|$ and $|\bar{s}_i|$ are the current and initial length of segment i. Since the input keypoints have almost identical distances between each other, the stretching energy is considered to be minimal already. λ_i and $\bar{\lambda}_i$ is the discrete twist between segment $i-1$ and i and the intrinsic twist. Twisting energy is difficult to measure because the twisting angle is hard to perceive. We assume both have zero value. Thus, in the approach proposed here, a DLO is considered smooth and physically realistic if it has low bending energy E_b. κ_i is the discretely estimated curvature at point i. $\kappa_i = 2tan(\psi_i/2)$, where ψ is the angle between two segments $i-1$ and i. Assuming no external contact and ignoring gravity, the DER model optimizes the total energy by applying internal force F_i at each keypoint. $F_i = -\frac{\partial E}{\partial q_i}$, where q_i is the generalized coordinates which include degrees of freedom of internal energies at the i^{th} keypoint.

First, a Savitzky-Golay filter is applied on the depths of the reconstructed keypoints [31] with a window size 17 to roughly smooth the DLO. Then, the DER model is applied to regularize all points and smooth the DLO. Two points at each end are fixed, assuming they are to be grasped (i.e. in a robot manipulation task). Our DER parameters have a similar order of magnitude to Lv's [25], where the DLO sectional radius is $r = 2 \times 10^{-4}m$ and the density is $7 \times 10^{-2}kg/m^3$. [2] These values were constants in all experiments. By keeping these two parameters fixed, adjusting the relative stiffness can be done by adjusting Young's modulus.

[2] These values are physically unrealistic used by Lv [25]. More realistic parameters could be estimated for the real DLO experiments, but it would still require an estimated compensating Young's modulus.

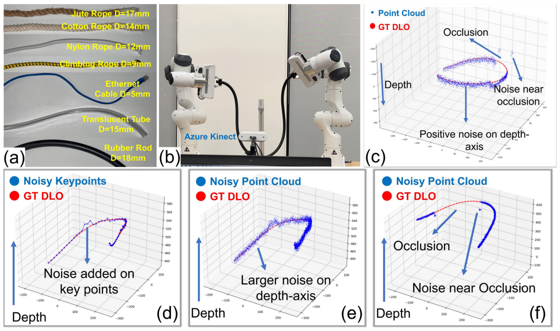

Figure 5: **Data Collection.** **(a)** Seven DLOs used in the experiments. **(b)** Real-world DLO data collection with two Franka arms and an Azure Kinect. **(c)** Synthetic DLO dataset generation, where the red points are the DLO ground-truth, and the blue points are synthetic point cloud points created by adding different types of noise to the DLO ground-truth. **(d)** Ground-truth DLO with keypoints and keypoints with added noise in the depth-axis. **(e)** Gaussian noise added in all directions, but larger in the depth-axis. **(f)** Random occlusion and noise added to the DLO point cloud near occlusion.

By initializing the positions of all points as described above, a single iteration of the DER fitting optimization is applied. One iteration provides a smoother DLO while keeping close enough to the measured 3D data (see Figure 9 for the visualization of the DER smoothing effect).

4 EXPERIMENTS

This section introduces the experiments and experimental setups, including data collection, 2D DLO segmentation, 3D DLO reconstruction, and DER-based smoothing.

4.1 Data Collection and Evaluation Metric

1. **Synthetic DLO datasets for quantitative evaluation.** The ground-truth of a real 3D DLO is hard to obtain, and it is hard to do quantitative evaluations without high-quality ground-truth. Thus, we followed a similar strategy to [6], i.e. to quantitatively evaluate reconstructed DLOs using a synthetic DLO dataset. We recorded 21 randomly manipulated DLOs with different stiffness and lengths of 60 keypoints in MuJoCo [32], recorded trajectories, and got aligned 2D masks and depth maps. Random segments of the synthetic DLOs are manually occluded, uniformly from 0% to 40%. 630 synthetic DLOs with masks and point cloud are used. Four types of noise were added to mimic different flying pixel issues on thin objects, as shown in the bottom row of Figure 2. The four types of noise are: (1) zero mean Gaussian noise with 6mm variance added to the keypoints in the depth direction (Figure 5 bottom left), (2) zero mean Gaussian noise with variance 2mm in non-depth directions, and 6mm in the depth direction applied to all DLO points (Figure 5 bottom middle), (3) positive uniform noise between 0mm to 12mm in the depth direction of random pixels (uniformly from 5% to 10%) on the DLO (Figure 5 c, (4) negative uniform noise between 6mm to 12mm in the depth direction added to DLO points near occlusion boundaries (within 1% of the total DLO length - see Fig 5 top right and bottom right). Noise (1) simulates the inaccurate depth of thin objects captured from depth sensors (see (h) in Figure 2). Noise (2) simulates noisy point cloud data. Noise (3) simulates flying pixels at the edge of the DLO (see (e) in Figure 2). Noise (4) simulates flying pixels at occlusions (see (g) in Figure 2). An example before and after adding depth-specific noise is shown in Figure 5.

2. **Real-world DLO datasets for testing.** A real dataset was collected from seven different DLOs: 1) a jute rope, 2) a cotton rope, 3) a nylon rope, 4) a climbing rope, 5) an Ethernet wire, 6) a translucent tube, and 7) a rubber rod. (See Figure 5 top left). These DLOs have different textures, color patterns, and radii. We collected at least 69 images of each DLO in different configurations with various occlusions and altogether 830 images for segmentation (94, 119, 205, 142, 69, 120, and 81 images of each DLO). The real-world dataset was reduced to 757 good-quality point cloud images (87, 116, 167, 131, 69, 116, and 71 images of each DLO). All images have an original resolution of 1920 × 1080 and were center-cropped to 1080 × 1080 pixels. The depth map has an original resolution of 640 × 576 collected by an Azure Kinect. We applied depth-to-color image alignment and center-

cropping to get the aligned depth map with the same resolution. We used unoccluded examples to get ground-truth point clouds and then introduced obstacles to create occlusion cases while keeping shapes the same.

3. **Evaluation metrics:** Intersection over Union score (IoU) evaluated the 2D DLO segmentation performance. Using only one metric to evaluate 3D DLO reconstruction does not comprehensively compare the performance of different reconstruction algorithms. Thus, four different 3D evaluation metrics were used:

a) D_{1bi} and D_1: D_{1bi} measures the overall similarity using the Chamfer distance $D_{1bi}(X, Y) = \sum_{x \in X} \min_{y \in Y} L_2(x, y)^2/|X| + \sum_{y \in Y} \min_{x \in X} L_2(x, y)^2/|Y|$, where X are the 3D reconstructed points and Y are the ground-truth points in the synthetic dataset or the unoccluded point cloud in the real-world dataset. The same interpretation of X and Y is used in the remaining formulas. D_1 measures the degree of fit: $D_1(X, Y) = \sum_{x \in X} \min_{y \in Y} L_2(x, y)^2/|X|$. A lower D_1 indicates the reconstructed DLO is closer to the unoccluded point cloud, although X may be incomplete.

b) D_2: When doing DLO grasping or DLO placement, D_2 estimates the worst-case distance that the reconstructed points can be away from a target grasping or placement position. We define $D_2(X, Y) = \max_{x \in X} \min_{y \in Y} L_2(x, y)$.

c) D_3 estimates the percentage of poorly reconstructed points by using a distance threshold T. $D_3(X, Y|T) = \sum_{x \in X} \mathbb{1}(\min_{y \in Y} L_2(x, y) > T)/|X|$ i.e. the percentage of reconstructed points whose distance to the closest ground-truth point (in the synthetic dataset or unoccluded cloud point in the real-world dataset) is larger than a certain threshold.

d) D_4: Physical smoothness is defined as $D_4(X) = \sum_{i=1}^{i=n-2} angle((X_{i+1} - X_i), (X_{i+2} - X_{i+1}))^2$ where $angle(A, B) = arccos(A \cdot B)/(||A|| \cdot ||B||))$. D_4 measures the sum of squares of angles between adjacent points.

As a false DLO connection or a partial reconstruction may miss a large area of the point cloud and lead to a significant increase in the distance metrics, Success Rate is used to measure the number of cases the DLO is reconstructed correctly. Only 'success' cases are used to calculate the four metrics. A DLO is reconstructed successfully when the metric $D_{1bi} < 50mm$, when using the synthetic dataset. In the case of the real-world dataset, a successful reconstruction requires 1) $D_1 < 25mm$, 2) $D_{1bi} - D_1 < 75mm$, and 3) $D_2 < 30mm$. The three criteria ensure that the reconstructed DLO satisfies: 1) the overall shape and position do not largely differ from the unoccluded point cloud, 2) the DLO points do not fit only a small part of the point cloud, and 3) a future manipulation does not fail because of a grasp or placement that was too far from the DLO.

4.2 2D DLO Segmentation

Four segmentation algorithms are compared to find which best performs the DLO segmentation task, with the results in Table 1. The four models are 1) **Wire-dataset**[10]: a segmentation model trained on a wire dataset proposed by Zanella et al., 2) **FastDLO**[12]: a real-time tracking model trained on a synthetic DLO dataset, 3) **SAM**: the segment anything model (SAM) [18] with the **Text** prompt *rope* as input. SAM itself does not allow a text prompt, so Grounded DINO [26] was used, which supports the text prompt for SAM. The box threshold and the text threshold are both 0.25, which satisfies all seven DLO segmentations. We only used *rope* rather than a more specific prompt for each image to avoid human supervision. 4) **Text+**, which post-processes the results from SAM using the methods from Section 3.1.

All models are evaluated on the real-world DLO dataset, including seven different DLOs (see Section 4.1). For the baseline models, we downsampled the resolution to 640×360 which is used in the original papers.

SAM was tested with different steps in Section 3.1, including **SAM Pixel**, which only removes components with few pixels, and **SAM Width**, which removes too-thin or too-thick masks based on SAM Pixel. A broad threshold range from 3 to 34 pixels was used *based on typical target sizes, distances, and camera parameters*. Four successful post-processing examples are shown in Figure 6.

Table 1 presents the results mentioned above:

1) The segmentation model trained on either the wire dataset or the rendered dataset does not perform well on other DLOs with different textures and color patterns.

2) The large vision model SAM with Grounding DINO computes a better segmentation result given the text prompt rope as input. SAM is convenient for segmenting unseen DLOs without any

Table 1: IoU scores ↑ and false positive rates (FPR) ↓ for the real-world DLO dataset. We use - to present the results of SAM Text+, as it has identical performance to SAM Text in the datasets with pure color backgrounds. As SAM achieves almost perfect results without false positive masks, post-processing provides no benefit.

Dataset	DLO-dataset based		SAM based	
	[10]	[12]	Text	Text+
Default	.052/.028	.227/.014	.775/.119	**.805/.002**
Grey	.021/.020	.491/.005	**.870/.001**	-
Green	.005/.020	.299/.011	**.872/.001**	-
White	.001/.021	.450/.009	**.867/.001**	-
Black	.028/.019	.424/.008	**.847/.001**	-

Table 2: IoU scores of SAM after each step on the datasets.

Dataset	SAM Text	SAM Pixel	SAM Width	SAM Text+
Full Dataset (830)	.775	.782	.794	**.805**
w/ occlusions (523)	.757	.761	.784	**.796**

model fine-tuning or prior knowledge. However, the false positive segmentations on the background or occlusions undermine the segmentation performance.

3) Post-processing based on morphological operations on SAM's segmentation masks improved the performance of SAM without any additional labeled training data measured by IoU scores. The post-processing significantly reduced false positive rates (FPR).

To reduce bias from the lab background, the real-world dataset's default background was replaced with four types of pure color (grey, green, white, and black). The last four columns of Table 1 show that the baseline model still cannot segment the DLO, supporting our opinions on the results in 1) and 2). Table 2 shows the performance of SAM on the full dataset. The **w/ occlusions** dataset only contains occlusions images, where SAM performs worse. Our post-processing improves DLO segmentation against occlusions.

SAM with Grounding DINO takes 8.2s to process an image on the resolution of 1920 × 1080 on average, and post-processing takes 0.6s to process the predicted mask at the resolution of 1080 × 1080 on an Nvidia RTX 3090.

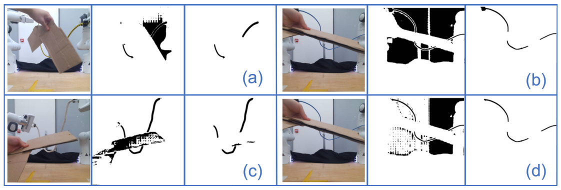

Figure 6: **Examples of segmentation post-processing**. Each example contains the input RGB image, the SAM segmented mask, and the post-processed mask.

4.3 3D DLO Reconstruction and DLO Smoothing

To evaluate the 3D geometric completion stage of the DLO reconstruction, the synthetic and the real-world DLO datasets were used with the four metrics from Section 4.1. The proposed 3D reconstruction algorithm was compared to the baseline reconstruction algorithm of Lv et al. [6], which is the State-of-the-Art. Two versions of the proposed method were evaluated: 1) **Rec**, the directly reconstructed DLO given as the output of Algorithm 1 and 2) **DER**, the smoothed DLO where Young's modulus was set to 40000 N/m^2. The **ground-truth** masks are used when comparing the 3D DLO reconstruction performance.

Table 3 shows the performance on the real-world and synthetic DLO datasets. The first row shows that the proposed model has a much lower value on all four metrics while maintaining a high success rate on the synthetic dataset. Several cases are visualized in Figure 7. When there is larger depth noise,

Table 3: Four metrics and success rate on the synthetic and real-world DLO datasets. D_3 has a threshold of $20mm$ on the synthetic dataset and $10mm$ on the real-world dataset.

DLO Type	$SuccessRate \uparrow$			$D_{1bi}(mm) \downarrow \& D_1(mm) \downarrow$			$D_2(mm) \downarrow$		
	Lv	Rec	DER	Lv	Rec	DER	Lv	Rec	DER
Synthetic (630)	.57	**.86**	.84	25.7/-	**13.1**/-	**13.1**/-	59.7	28.1	**27.6**
Real (757)	.21	.66	**.67**	**16.9**/5.5	19.6/**3.0**	20.4/3.4	18.1	**10.5**	11.0
Real 75% (464)	-	-	-	32.5/14.3	**18.3**/**4.7**	19.0/**4.7**	61.6	**19.7**	20.2

DLO Type	$SuccessRate \uparrow$			$D_3 \downarrow$			$D_4 \downarrow$		
	Lv	Rec	DER	Lv	Rec	DER	Lv	Rec	DER
Synthetic (630)	.57	**.86**	.84	.165	**.054**	**.054**	1.9	4.6	**0.4**
Real (757)	.21	.66	**.67**	.101	**.050**	**.050**	1.5	1.8	**0.3**
Real 75% (464)	-	-	-	.374	**.091**	.092	2.6	1.9	**0.4**

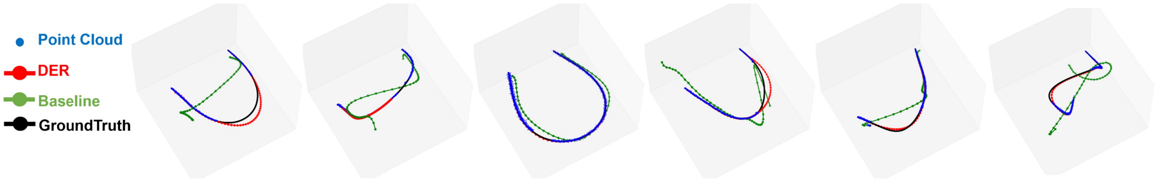

Figure 7: 3D reconstruction visualization on the synthetic DLO datasets. Our result, red curves overlaying on ground-truth curves in most cases, outperforms the baseline model with more correct connection and fitting.

the baseline model fails to estimate the correct DLO points that fit the point cloud, and performance also deteriorates with occlusions.

The 3D reconstruction experiments used the unoccluded DLO point cloud as the ground-truth. All depth pixels outside the workspace were cropped, which has a depth range between 0.2m to 0.8m, though much depth noise remains close to the DLO due to the problems illustrated in the bottom row of Figure 2. The ground-truth 3D point cloud does not describe the DLO well due to depth noise and missing depth pixels. Visualizations of results from the real-world datasets are seen in Figure 8.

Table 3 shows that the proposed models have a much higher success rate on real-world DLO reconstructions. Performance on the rubber rod was much worse than the other DLOs because the black rubber rod has a more reflective surface than other DLOs, which leads to point clouds of poor quality, even without occlusions. This increases the difficulty of both DLO reconstruction and evaluation. Figure 8 shows one case where the point cloud is incomplete even without occlusions, and successful reconstruction of the missing part of the point cloud increases all distance metrics.

Table 3 evaluates all models with D_1 and D_{1bi}:

1) The proposed approaches have a relatively low D_1 distance compared to the baseline, and the

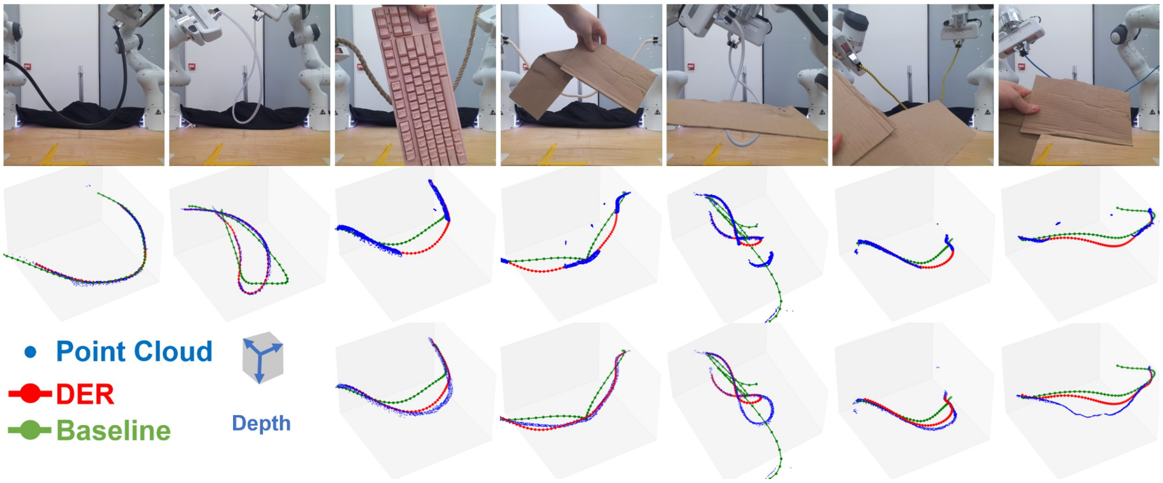

Figure 8: 3D reconstruction visualization on the real-world DLO dataset. Considering we do not have ground-truth keypoints, we overlay reconstructed DLOs on unoccluded point cloud in the last row to show connection and fitness.

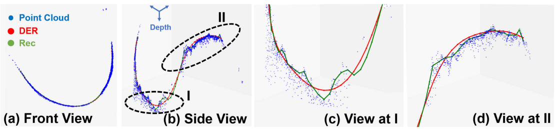

Figure 9: An example of the DER smoothed DLO from two views.

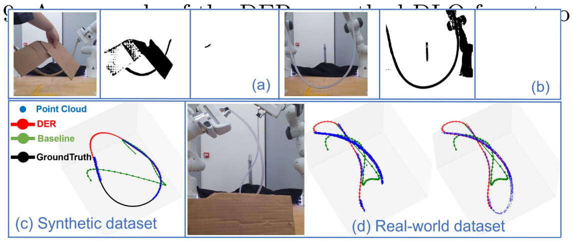

Figure 10: Failure cases; (a), (b) on post-processing where the DLO mask is removed. (c), (d) on 3D reconstruction where the connection is incorrect.

direct reconstruction method performs the best. This can be foreseen because the reconstructed points are all from the point cloud.

2) The baseline method performs well on D_{1bi} compared to the proposed methods. However, it has a much lower success rate (0.21 compared to 0.66 and 0.67 of the proposed methods). Because only successful cases are calculated in D_{1bi}, the baseline has a much lower point distance on the more strictly selected successful cases.

3) The method using the DER to smooth the DLO performs slightly worse, because smoothing reconstructed points makes some points not fit well. Figure 9 shows an example of the 3D points, the reconstructed DLO, and the smoothed DLO. The smoothed DLO is clearly better.

Table 3 shows that the proposed methods have lower D_2 distances (maxi-min error) on the real-world dataset. Thus, the worst grasping or placement position can be within a bounded error, even if the grasp position is unluckily chosen.

D_3 shows that both the direct reconstruction and DER methods have the lowest percentage of reconstructed points whose distance is larger than $10mm$ to the closest point in the unoccluded point cloud.

D_4 shows that the DER-based smoothing produces the physically smoothest reconstructions with a slight sacrifice on the closeness of fit, as measured by D_1, D_{1bi}, and D_2. One reason is that the DLOs in the real world usually have different levels of plastic deformation (see the DLOs in Figure 5 a). But the DER model assumes the DLO has no plastic deformation, and its default shape should be straight.

In the last row in Table 3, a second comparison was based on selecting the DLO images with the lowest 75% D_{1bi} from each of the three methods and then intersecting the three subsets to produce a reduced test subset. This practice considers non-outlier failure cases to avoid unfair comparisons. In this case, both proposed methods outperform the baseline, which supports the claim that the good performance of the baseline on D_{1bi} is a consequence of its reduced test sample selection and benefits from samples of good quality.

Some failure cases are: 1) A false connection across large occlusions (see the first figure of Figure 7 and the last figure of Figure 8). 2) A poor reconstruction when there is a large occlusion on a non-straight part (see 6th column of Figure 8). See also failure cases from both 2D segmentation and 3D reconstruction in Figure 10.

On average, 3D reconstruction takes 0.8s in total, and DER physical smoothing takes less than 0.1s on Intel i9-10850K.

5 LIMITATIONS and CONCLUSIONS

Limitations: Our pipeline takes around 10 seconds to process an image of 1080×1080. Though we do not aim to provide real-time perception, improving efficiency remains challenging, especially in DLO segmentation. Here, we focused on the scenarios with one single DLO in the scene. Another future work is to generalize to cases with multiple DLOs.

Conclusion: This paper proposes a 3D DLO perception pipeline, including 2D segmentation, 3D reconstruction, and smoothing based on DER, the physical model. Improved DLO segmentation performance was achieved by using a large vision segmentation model (SAM) and post-processing the masks by further eliminating false positive segmentations to improve the segmentation performance without additional training data. The proposed 3D reconstruction method, by using geometric completion, can reduce the depth noise of the DLOs. Finally, the DER method physically smooths the reconstructed DLOs. Experiments on both a synthetic and a real-world dataset of 7 different DLOs demonstrate that the proposed methods outperform previous algorithms on both 2D segmentation and 3D reconstruction. We believe the proposed 3D DLO perception pipeline can provide a good DLO perception initialization suitable for downstream robotic tasks on DLOs, such as DLO tracking and manipulation.

References

[1] J. Zhu *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE RAM*, 2022.

[2] J. Sanchez *et al.*, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *IJRR*, 2018.

[3] H. Yin *et al.*, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, 2021.

[4] T. Tang *et al.*, "Track deformable objects from point clouds with structure preserved registration," *IJRR*, 2018.

[5] Y. Wang *et al.*, "Tracking partially-occluded deformable objects while enforcing geometric constraints," in *ICRA*. IEEE, 2021.

[6] K. Lv *et al.*, "Learning to occlusion-robustly estimate 3-d states of deformable linear objects from single-frame point clouds," *arXiv preprint arXiv:2210.01433*, 2022.

[7] M. Yu *et al.*, "A coarse-to-fine framework for dual-arm manipulation of deformable linear objects with whole-body obstacle avoidance," in *2023 ICRA*. IEEE, 2023, pp. 10 153–10 159.

[8] J. Xiang *et al.*, "Trackdlo: Tracking deformable linear objects under occlusion with motion coherence," *IEEE RA-L*, 2023.

[9] W. Zhang *et al.*, "Deformable linear object prediction using locally linear latent dynamics," *arXiv preprint arXiv:2103.14184*, 2021.

[10] R. Zanella *et al.*, "Auto-generated wires dataset for semantic segmentation with domain-independence," in *ICCCR*. IEEE, 2021.

[11] B. Thananjeyan *et al.*, "All you need is luv: Unsupervised collection of labeled images using invisible uv fluorescent indicators," *arXiv preprint arXiv:2203.04566*, 2022.

[12] A. Caporali *et al.*, "Fastdlo: Fast deformable linear objects instance segmentation," *IEEE RA-L*, vol. 7, no. 4, pp. 9075–9082, 2022.

[13] P. Zhou *et al.*, "Lasesom: A latent and semantic representation framework for soft object manipulation," *IEEE RA-L*, 2021.

[14] M. Yu *et al.*, "Shape control of deformable linear objects with offline and online learning of local linear deformation models," *arXiv preprint arXiv:2109.11091*, 2021.

[15] Y. Yang *et al.*, "Learning to propagate interaction effects for modeling deformable linear objects dynamics," in *ICRA*. IEEE, 2021.

[16] A. Keipour *et al.*, "Deformable one-dimensional object detection for routing and manipulation," *arXiv preprint arXiv:2201.06775*, 2022.

[17] P. Kicki *et al.*, "Dloftbs–fast tracking of deformable linear objects with b-splines," *arXiv preprint arXiv:2302.13694*, 2023.

[18] A. Kirillov *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[19] X. Zhang *et al.*, "Channel attention in lidar-camera fusion for lane line segmentation," *Pattern Recognition*, vol. 118, p. 108020, 2021.

[20] A. Dietsche *et al.*, "Powerline tracking with event cameras," in *IROS*. IEEE, 2021, pp. 6990–6997.

[21] J. Yuan *et al.*, "Image feature based gps trace filtering for road network generation and road segmentation," *Machine Vision and Applications*, vol. 27, no. 1, pp. 1–12, 2016.

[22] Y. Wang *et al.*, "Deep distance transform for tubular structure segmentation in ct scans," in *CVPR*, 2020, pp. 3833–3842.

[23] A. Caporali *et al.*, "Ariadne+: Deep learning-based augmented framework for the instance segmentation of wires," *IEEE Transactions on Industrial Informatics*, 2022.

[24] M. Bergou *et al.*, "Discrete elastic rods," in *SIGGRAPH*, 2008.

[25] N. Lv *et al.*, "Dynamic modeling and control of deformable linear objects for single-arm and dual-arm robot manipulations," *IEEE T-RO*, 2022.

[26] S. Liu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[27] S. Shao *et al.*, "Objects365: A large-scale, high-quality dataset for object detection," in *Proc IEEE/CVF Int Conf on Computer Vision*, 2019, pp. 8430–8439.

[28] A. Choi *et al.*, "mbest: Realtime deformable linear object detection through minimal bending energy skeleton pixel traversals," *arXiv preprint arXiv:2302.09444*, 2023.

[29] M. Yan *et al.*, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE RA-L*, 2020.

[30] M. Hofer *et al.*, "Energy-minimizing splines in manifolds," in *SIGGRAPH*, 2004, pp. 284–293.

[31] R. W. Schafer, "What is a savitzky-golay filter?[lecture notes]," *IEEE Signal processing magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[32] E. Todorov *et al.*, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ IROS*. IEEE, 2012, pp. 5026–5033.