2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Deformable Objects Perception is Just a Few Clicks Away – Dense Annotations from Sparse Inputs

Alessio Caporali, Kevin Galassi, Matteo Pantano, Gianluca Palli

***Abstract*—Deformable Objects (DOs), e.g. clothes, garments, cables, wires, and ropes, are pervasive in our everyday environment. Despite their importance and widespread presence, many limitations exist when deploying robotic systems to interact with DOs. One source of challenges arises from their complex perception. Deep learning algorithms can address these issues; however, extensive training data is usually required. This paper introduces a method for efficiently labeling DOs in images at the pixel level, starting from sparse annotations of key points. The method allows for the generation of a real-world dataset of DO images for segmentation purposes with minimal human effort. The approach comprises three main steps. First, a set of images is collected by a camera-equipped robotic arm. Second, a user performs sparse annotation via key points on just one image from the collected set. Third, the initial sparse annotations are converted into dense labels ready for segmentation tasks by leveraging a foundation model in zero-shot settings. Validation of the method on three different sets of DOs, comprising cloth and rope-like objects, showcases its practicality and efficiency. Consequently, the proposed method lays the groundwork for easy DO labeling and the seamless integration of deep learning perception of DOs into robotic agents.**

***Index Terms*—Deformable Objects, Semantic Segmentation, Dataset Generation, Deep Learning, Garment Perception, Cloth**

I. Introduction

Deformable Objects (DOs) refer to objects with the ability to change their shape when subjected to external forces. They are commonly encountered in everyday life, such as clothes and garments, which are commonly referred to as Deformable Planar Objects (DPOs), or cables, wires, and ropes, known as Deformable Linear Objects (DLOs). These objects are also prevalent in various fields, including the medical [1], agricultural [2], and industrial domains [3].

The problem of sensing and manipulating DOs is an emerging research topic in robotics [4]. Since DOs are ubiquitous in our daily lives, the ability to perceive and manipulate them is a crucial skill for robots to possess [5]. For instance, robots can provide valuable assistance in elderly care by assisting with tasks such as dressing and handling textile objects [6].

Alessio Caporali, Kevin Galassi, and Gianluca Palli are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, IT-40136 Bologna, Italy.

Matteo Pantano is with the Department of Electrical and Computer Engineering, Technical University of Munich, D-80333 Munich, Germany.

This work was partially supported by the Horizon Europe project *IntelliMan - AI-Powered Manipulation System for Advanced Robotic Service, Manufacturing and Prosthetics* [grant number 101070136].

Corresponding author: *alessio.caporali@unibo.it*

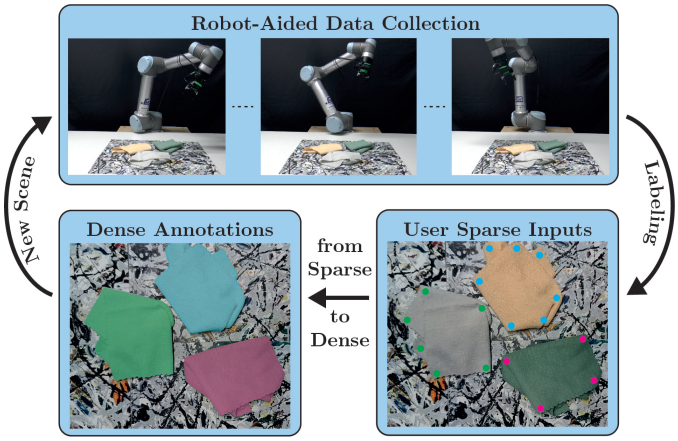


Fig. 1: Schematic of the proposed method: The robot is equipped with an eye-in-hand camera to effortlessly collect several camera samples. The user sparsely annotates each object in only one image, and dense annotations are extracted for all images using a pre-trained foundation model.

The perception of DOs poses challenges due to their inherent deformability, which makes their shape unpredictable, as well as the limited (or possibly lack of) relevant features to be used in common computer vision approaches [7]. To address these challenges, new perception methodologies based on deep learning are crucial, particularly concerning the segmentation of DOs. However, these approaches require training data. The size and quality of datasets greatly affect the performance of existing data-driven approaches, particularly in the DOs domain [8], making the development of efficient data collection and labeling procedures desirable.

In previous work, we introduced *DLO-WSL*, a method for labeling DLOs such as cables and wires with minimal effort using a spatial sensor and an eye-in-hand robot camera [7]. However, the approach relied on knowledge of the target DLO (e.g., diameter) and a specifically designed learned label-tuning algorithm for error correction. Therefore, *DLO-WSL* is not readily applicable to other DOs and is subject to domain shift problems when dealing with DLOs with quite different textures, such as ropes. In this paper, we focus on eliminating these constraints to convert sparse annotations into dense masks in the most general and versatile manner possible. By harnessing the capabilities of a pre-trained foundational model [9], specifically the Segment Anything Model (SAM) [10], we can convert any sparse annotation into a dense one without requiring fine-tuning steps or domain-specific knowledge.

Additionally, the utilization of an eye-in-hand camera robot allows us to expand the set of samples collected with just one annotation. This enhances the method's portability and efficiency by simplifying the approach and making it more cost-effective, thus increasing the likelihood of adoption by industrial practitioners.

To summarize, the main contributions of this work are:

- Generation of a dataset for domain-specific segmentation of DOs with minimal human intervention.
- Exploitation of pre-trained foundation models for dense mask annotation.
- Comprehensive experimental validation of the method across different classes of DOs and application of the obtained dataset for a real-case downstream segmentation task.

II. Related Works

A. Deformable Objects Perception

1) Deformable Linear Objects: The perception of DLOs is commonly achieved through either vision-based [7], [11], [12] or tactile-based sensors [13]. The former is preferred owing to the availability of diverse sensors and cameras that seamlessly integrate into robotic systems [14]. The latter is required for tight spaces and occlusions, where vision-based perception struggles.

Research efforts in the vision-based domain have concentrated on employing data-driven methods for semantic segmentation, e.g. in [7], [15]–[22]. Dataset generation is the primary challenge in deep learning due to the labor-intensive process of gathering and labeling extensive datasets for training. Many works resort to manual annotation, e.g. [17]–[20], resulting in a tedious, inaccurate, and non-scalable process. As tasks become more visually complex, such as DLO segmentation, annotation becomes slower and more challenging. To address this, some studies explore dataset-generation methods with minimal or no human intervention [7], [15], [21]. However, approaches like those in [21] and [15] face limitations, including susceptibility to incorrect labels due to color separation. Differently, the method proposed in [7] can suffer from domain gap problems and requires object-specific knowledge.

The semantic segmentation of DLOs is accomplished using various off-the-shelf deep learning models, including UNet in [18] and [21], FCN in [17], and DeepLabV3+ in [7], [15], [18]. Additionally, a custom Convolutional Neural Network (CNN) architecture with an encoder-decoder scheme is proposed in [20] and [19]. In [22], the pre-trained Segment Anything Model [10] is used for DLO segmentation without specific fine-tuning, showing its potential. However, satisfactory results require additional post-processing steps [22]. Comparisons between real-world and synthetic datasets are conducted in [18] and [7]. Both evaluate synthetic datasets against the *electrical wires dataset* by [15], which combines real DLO images with synthetic backgrounds. The findings suggest that synthetic images can serve as a viable alternative for DLO segmentation. Additionally, a combination of synthetic and real-world images improves segmentation performance compared to using synthetic images alone [7].

Concerning the instance segmentation task, the approaches presented in [7] and [16] aim to obtain instance-wise masks for DLOs using fully synthetic methods. Additionally, they employ state-of-the-art instance segmentation models. However, weaker performances, especially when multiple DLOs intersect, are observed compared to the semantic segmentation task. This emphasizes the need for further research in developing dataset generation techniques tailored for fully deep-learning-based instance segmentation methods for DLOs.

2) Deformable Planar Objects: Compared to DLOs, the perception of DPOs presents an even increased level of difficulty due to the possible severe occlusions of object parts which makes the estimation of the object state quite complex [23]. The literature of DPOs research has investigated different aspects, like cloth classification and segmentation [24], wrinkle detection [25], key points (e.g., corners) or boundaries detection [26] and full state estimation [27].

Data-driven methods are a commonly employed strategy to address DO perception. However, since simulated environments are usually employed, the sim-to-real gap needs to be addressed. Pseudo-labels of real-world data are investigated for fine-tuning simulation-trained models in the same work [27]. In this paper, by leveraging the proposed labeling approach on real-world data only, the need for fine-tuning steps is avoided, and learning is performed directly from high-quality real images.

B. Image Labeling and Sparse Annotation

Efficient generation of ground truth segmentation masks is crucial for training data-driven models. As highlighted in the context of DLOs perception, manual annotation, while effective, becomes burdensome for large datasets, prompting the exploration of alternative methods to simplify the annotation process.

In the literature, various sources such as image labels, point clicks, bounding boxes, scribbles, and saliency detection have been proposed [28]–[32]. Interactive segmentation methods, like iterative correction based on bounding boxes [33] or user-labeled superpixels [34], aim to refine coarse annotations. Another strategy involves the automatic adjustment of coarse labels using specific knowledge [35] or gradient guidance [36]. RITM [37] leverages pre-trained networks for efficient user-assisted segmentation. These methods, while effective for single images, face challenges in scaling to large datasets.

Other works leverage SAM for image labeling. For instance, [38] utilizes SAM in an interactive annotation process involving satellite imagery, while [39] explores SAM's application in the medical domain. In contrast, we employ SAM in a minimal-effort annotation process, enabled by the exploitation of a camera-equipped robotic arm.

III. Method

The proposed approach is based on the concept that by employing an eye-in-hand camera, multiple images can be

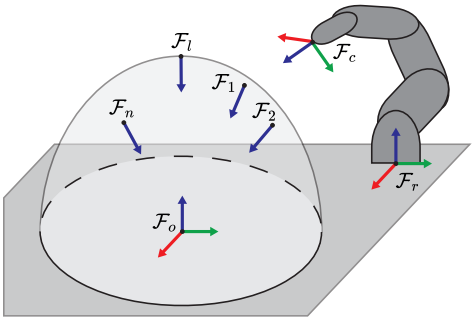


Fig. 2: Schematic view of the robotic setup, ellipsoidal trajectory, and main reference frames utilized.

labeled by annotating a single image only. To implement this approach, especially considering DOs in real-world scenarios, two key components are exploited: a sparse key points annotation approach employing a simple interface, as detailed in Sec.III-A, and the use of SAM [10] for dense mask generation, as discussed in Sec.III-B.

A. Dataset Collection and Sparse Key-points Input

1) Data Collection: To collect the set of images, knowledge of the images and the camera's position in the world coordinate system is essential [7], [40]. This is accomplished by using a calibrated 2D RGB camera mounted on the flange of a robotic arm in an eye-in-hand setup. With this information, an ellipsoidal robot trajectory is executed to collect visual samples of the DOs, ensuring that the object remains at the trajectory center while the camera is inward-facing. This is achieved by executing a trajectory described in Eq. 1.

$$\begin{cases} x = a\sin\theta\sin\phi + x_0 \\ y = -b\sin\theta\sin\theta + y_0 \\ z = c\cos\theta + z_0 \end{cases} \tag{1}$$

Here, x, y, z represent a specific point on the trajectory, θ denotes the elevation angle, ϕ represents the heading angle, a, b, c are the ellipsoid parameters, and x_0, y_0, z_0 indicate the initial position coordinates.

Given the point $p = [x, y, z]^T$, an orientation is computed to fully characterize the camera pose. By considering the ellipsoid center $p_0 = [x_0, y_0, z_0]^T$, the direction vector pointing toward the inside from p is computed as $p_0 - p$. Therefore, by also considering another vector, such as $v = [0, 1, 0]^T$, the pose for point p is computed using the cross product of these vectors.

An illustration of the ellipsoidal trajectory with several reference frames (e.g., $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n$) is shown in Fig. 2. For clarity, only the z-axis is shown in several frames. The camera frame is denoted as $\mathcal{F}_c$.

2) Key Points-based User Sparse Annotation: Next, a top-view perspective image from the collected dataset is used to generate sparse labels for the entire dataset ($\mathcal{F}_l$ in Fig. 2).

Users are then instructed to trace a sequence of key points along the object's shape, aided by an intuitive visualization to facilitate the labeling process. The labeling methodology differs slightly between DLOs and DPOs. For DLOs, key points are roughly traced along the centerline, while for DPOs, they are marked along the interior border, following the object's perimeter. This approach ensures an intuitive and efficient labeling process for both DLOs and DPOs.

Subsequently, with knowledge of the camera pose and the specific camera perspective parallel to the working plane, each input key point is projected into Cartesian space as outlined by Eq. 2

$$\begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = {}^cT_r \begin{bmatrix} \frac{u_i - c_x}{f_x} \\ \frac{v_i - c_y}{f_y} \\ 1 \\ 1 \end{bmatrix}, \forall i \tag{2}$$

Where u_i and v_i are the labeled pixels, c_x, c_y, f_x, and f_y are the camera parameters obtained from the camera intrinsics matrix, ${}^cT_r$ is the extrinsic matrix of the camera obtained by knowing the camera position in the world coordinate frame, and x_i, y_i, and z_i are the world coordinates of the labeled point.

B. Dataset Labeling via Foundation Models

The key points annotated by the user and converted into world coordinates (as described in Sec.III-A2) are utilized to fully label the dataset collected by leveraging the ellipsoidal trajectory detailed in Sec.III-A1. Therefore, these points are projected onto the designated image plane (that needs to be labeled) and utilized to initiate SAM [10], the pre-trained foundation model employed for dense pixel-wise labeling of images. A schematic overview of the dataset labeling approach is depicted in Fig. 3.

1) Sparse Inputs Projection: First, the 3D points computed in Sec.III-A2 are projected onto the specific image plane. Indeed, with the world coordinates for the labeled points established, each captured sample from Sec.III-A1 can be labeled without further user input. Specifically, by employing the inverse relation to that utilized in Eq. 2, the key points provided by the user are projected back onto the required image. In other words, the manual generation of key points for each image in the dataset is replaced by the camera-equipped robot and its associated camera-robot transformation, enabling seamless conversion between image coordinates and world coordinates.

2) Transforming Sparse Inputs to Dense Labels: Given the 2D key points, dense masks are directly generated by leveraging SAM.

The SAM network consists of three primary components: 1) an RGB image encoder, which utilizes a ViT transformer; 2) a prompt encoder capable of accepting bounding boxes or key points as input; and 3) a mask decoder responsible for computing the output mask based on the embedded image and prompt.

In this paper, we apply the readily available pre-trained weights of SAM, thereby avoiding costly and unnecessary fine-tuning procedures. Currently, SAM does not support

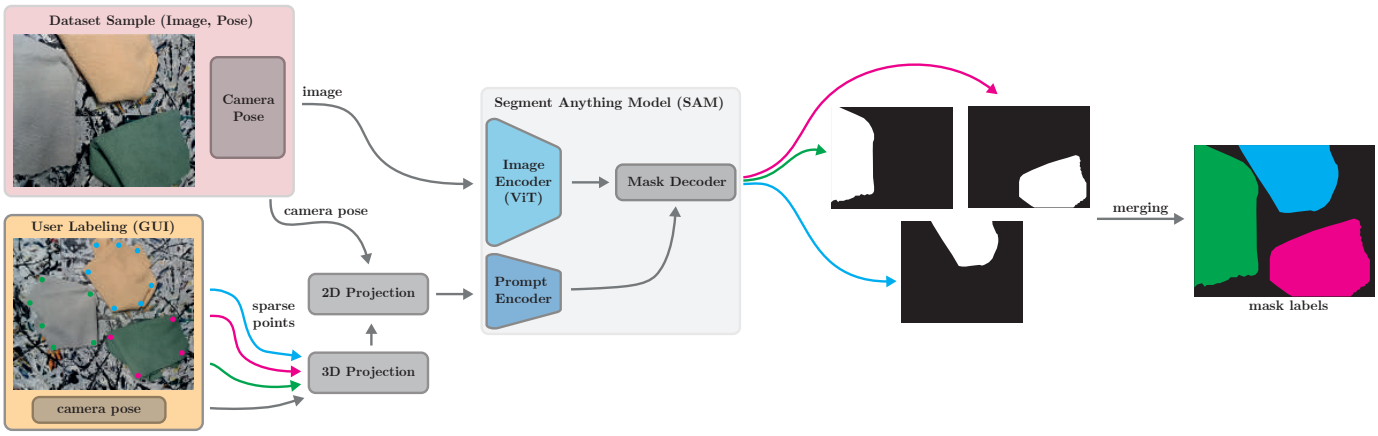


Fig. 3: Dataset labeling using the Segment Anything Model (SAM) involves projecting the user's sparse labels into world coordinates. Subsequently, for each sample in the dataset, SAM is prompted with the projected points corresponding to each object of interest. Finally, merging is performed to obtain the final labels.

multi-object prompting. Therefore, when labeling multiple objects is necessary, SAM must be prompted separately for each. However, the most computationally intensive task, i.e. the image embedding via the vision transformer, needs to be performed only once and can be saved for subsequent use with different prompts. In contrast, the prompt encoder and mask decoder are relatively small and efficient models.

In practice, the raw unthresholded prediction is obtained by SAM by setting an explicit flag. When labeling D objects, such as $D = 3$ as illustrated in Fig. 3, each object undergoes specific prompting, and the resulting masks are concatenated. Consequently, an overall mask of dimensions $H \times W \times D$ is produced, where H and W represent the height and width of the image, respectively. An additional empty mask, filled with zeros, is appended to accommodate the background "class". Subsequently, the softmax activation function is applied to the concatenated masks to derive probability values across the dimension D. Finally, the merged mask is obtained by executing the *argmax* function along the last dimension, as depicted on the right side of Fig. 3.

IV. Experiments

The method's evaluation is presented in two parts: initially, a comparison between sparse labeling and a baseline approach is conducted to underscore the advantages of the proposed method. Subsequently, by leveraging the proposed pipeline, a dataset of DOs is generated and utilized to optimize a segmentation network, thereby validating the method's effectiveness for downstream perception tasks.

The first aspect involves a user test, elaborated on in Secs. IV-B and IV-C. Details regarding the segmentation task are outlined in Sec. IV-D. Both experiments utilize the same set of data samples for training, validation, and testing, the specifics of which are provided in Sec. IV-A.

A. Data Collection

The proposed approach is validated through experimentation involving various types of DOs, including cloth-like materials and rope-like ones. The cloth materials are further categorized into two groups: *Group A* consists of three soft cloths with uniform colors, while *Group B* comprises two soft cloths with complex colors and textures. Additionally, *Group C* is composed of three different ropes of varying colors and diameters. These sets of DOs are illustrated in Fig. 4.

(a) Group A (b) Group B (c) Group C

Fig. 4: The sets of deformable objects employed in the experiments consist of three uniform-colored soft clothes, two textured soft clothes, and three ropes with different textures and diameters.

The data samples are acquired using the robotic configuration depicted in Fig. 1, comprising a UR5 robot manufactured by Universal Robots equipped with an eye-in-hand OAK-1 camera provided by Luxonis. The camera resolution is set at 1080×1920 pixels, and it has been both intrinsically and extrinsically calibrated relative to the robot flange.

The ellipsoidal trajectory detailed in Sec. III-A is executed with the following parameters selected taking into consideration the robot workspace and the camera field of view, specifically: $a = 0.35$, $b = 0.35$, $c = 0.35$, elevation angle steps 5, maximum elevation angle 50, heading angle steps 5.

To train the segmentation network, six different backgrounds are utilized to generate the data scenarios, evenly distributed between tier-1 and tier-2. The three tier-1 backgrounds consist of uniform colors without complex shapes or distracting elements. In contrast, the remaining three tier-2 backgrounds feature intricate and chaotic shapes, resulting in a visually complex appearance. A video demonstrating the robot collecting dataset samples across the various backgrounds is provided as supplementary material.

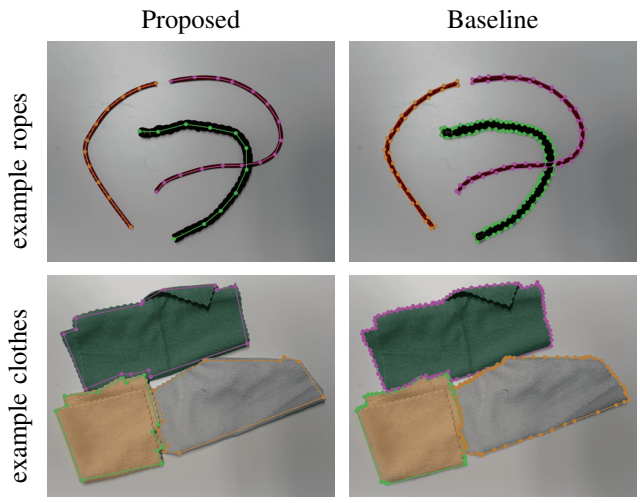
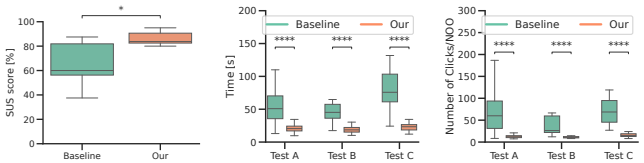


Fig. 5: Visualization of sparse user annotations on rope or cloth-like objects, performed using either the proposed method or the baseline approach.



(a) SUS (b) Time (c) Number of Clicks

Fig. 6: User test results concerning System Usability Scale (a), normalized labeling time per object (b) and normalized number of clicks per object (c). NOO stands for number of objects.

In total, the robot collects 30 data samples from each scene using the ellipsoidal trajectory. Thus, across all six backgrounds, a dataset of 180 samples is generated, requiring sparse annotation for only 6 samples, which constitutes 3% of the entire set.

To conduct testing, four additional backgrounds are exclusively utilized to gather test data samples, which are then employed to assess the method's performance in labeling and segmentation. Specifically, 5 samples per background are selected for testing purposes, resulting in 20 test samples for each group, denoted as *Test A*, *Test B*, and *Test C*. Finally, each test image is precisely annotated by a human expert to obtain accurate pixel-level ground truth data.

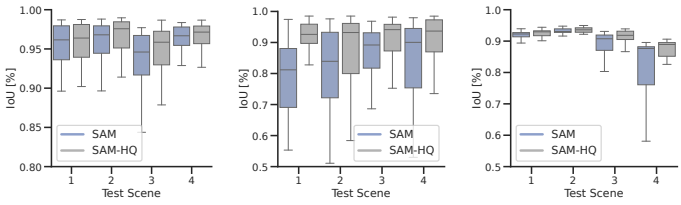
B. Sparse Labeling User Test

A user test was designed with a balanced randomized order of two subsequent interactions to evaluate the impact of the proposed labeling approach. The labeling interface comprises a full-screen visualization of the image to annotate. Users are prompted to draw a polygonal shape around the object of interest, primarily using a mouse and occasionally employing a few keyboard keys for functions such as deleting or saving. Throughout the labeling process, the currently drawn polygonal shape is displayed to simplify the understanding of the process. Due to the intrinsic advantages of the proposed approach, users are asked to draw a rough shape following the interior borders of the objects (see Sec.III-A2). In contrast, as a baseline approach, simple labeling of the object following its contour with as much precision as possible is employed. To better highlight the difference between the two labeling approaches, a comparative view of the labeling performed by one user is provided in Fig. 5.

For these comparisons, users were asked to label four images (one image per test object background) for each test set, resulting in a total of 12 images, using the two different methods. After each interaction, usability was assessed using the System Usability Scale (SUS) [41]. Additionally, the number of clicks (NoC) and the total time taken to complete the labeling task were recorded.

A total of 10 users with low to medium experience levels in labeling techniques, aged 25.3 ± 2.90 years participated in the study. Experiments were conducted in compliance with the Declaration of Helsinki, and all participants signed an informed consent form. All of them performed the test correctly, and no data were discarded.

The results of the user test concerning SUS, normalized time (time/NOO) and number of clicks (NoC/NOO) for each labeled object are shown in Fig. 6. Concerning usability, SUS scored a higher evaluation of the proposed method when compared with the baseline. More precisely, our method received a SUS score of 85.75 ± 5.53 whereas the baseline scored 65.75 ± 16.28. This proved to be statistical significant when the Mann-Whitney U test was applied as long the preconditions for the t-test did not hold. Therefore, it is possible to deduce that our method proves to be more usable compared to the baseline. Concerning the normalized labeling time for each labeled object, our method proved to allow for faster labeling. Specifically, across the test samples, we observed 21.7 ± 7.6 s/NOO, 19.8 ± 7.3 s/NOO and 23.3 ± 6.8 s/NOO for *Test A*, *B*, and *C* with the proposed method. Instead, the baseline methodology requires 55.2 ± 29.1 s/NOO, 45.3 ± 18.7 s/NOO and 80.6 ± 38.6 s/NOO respectively. This proved to be statistical significant when the Mann-Whitney U test was applied as long the preconditions for the t-test did not hold. Therefore, it is possible to deduce that the proposed labeling method allows for faster labeling. Notably, the proposed sparse labeling method allows for about 2.5X and 2.2X gain in time on average for cloth-like objects and 3.8X for rope-like objects. Concerning the number of clicks, a pattern similar to that of time is observed. More precisely, a decrease in the required number of clicks is observed. Indeed, across the test samples, we observed 13.3 ± 3.9 NoC/NOO, 11.6 ± 3.4 NoC/NOO and 18.1 ± 9.5 NoC/NOO for *Test A*, *B*, and *C* with the proposed method. Instead, the baseline methodology requires 79.3 ± 66.9 NoC/NOO, 63.23 ± 72.7 NoC/NOO and 82.17 ± 48.16 NoC/NOO respectively. This proved to be statistical significant when the Mann-Whitney U test was applied as long the preconditions for the t-test did not hold. Therefore, it is possible to claim that the proposed method can reduce NoC/NOO by factors of 6.0X, 5.5X, and



(a) Test A (b) Test B (c) Test C

Fig. 7: Quality of obtained dense labels on the different test sets when employing SAM or SAM-HQ for dense annotation. Test scenes 1 to 4 denote the specific combination of test background and test objects.

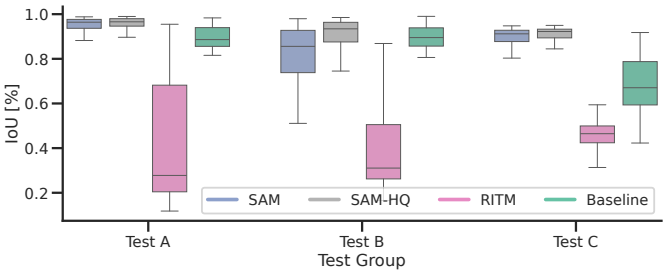


Fig. 8: Comparison between dense label generation employing the sparse or *Baseline* points as input. Sparse points used by: SAM, SAM-HQ and RITM.

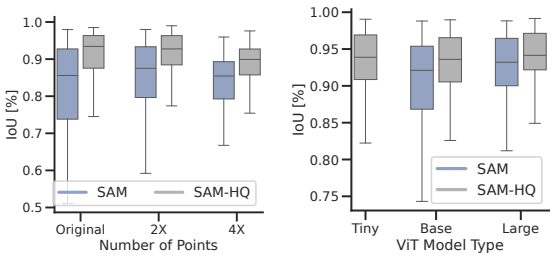
4.5X correspondingly for *Test A*, *B*, and *C*. Moreover, the proposed method allows for a more diverse user group to perform equally well as the standard deviation of NoC/NOO is drastically reduced if compared with the baseline approach.

C. Dense Labels Quality

The quality of the obtained dense labels is evaluated across five images for each scenario, totaling 20 samples per DOs group, as detailed in Sec. IV-A. Starting from the key points annotated by each user (sparse points), the input is propagated to the other five images of the scenario following the proposed pipeline (Sec. III). The dense annotation starting from the sparse input points is performed using SAM [10]. As alternative approaches, the SAM-HQ model [42] and RITM [37] are also investigated. Specifically, all three methods require only the set of points as input to produce the segmentation mask. Therefore, the same set of user input points is provided to all methods, allowing a comparison of their performances.

As a reference, the Baseline points annotated by each user, as described in Sec. IV-B, are employed to compute a dense label using the same projection approach but without the aid provided by SAM, SAM-HQ, or RITM.

The intersection over union (IoU) score is employed as a metric for comparing the annotated masks to the ground truth data. The IoU is defined as $\frac{|M \cap M_{gt}|}{|M \cup M_{gt}|}$, where M is the mask obtained by the proposed method and M_{gt} is the ground truth mask. Therefore, the IoU is computed for each object in the scene, and the average is calculated. The results comparing the accuracy of SAM and SAM-HQ are shown in Fig. 7.



(a) Effects number of points (b) Effects model size

Fig. 9: (a) Effects on the number of prompt points on Test B. (b) Comparison of different model sizes across the test set.

Backbone	*Test A*		*Test B*		*Test C*	
	IoU	Dice	IoU	Dice	IoU	Dice
ResNet-50	0.902	0.940	0.938	0.967	0.868	0.927
ResNet-101	0.924	0.954	0.940	0.968	0.833	0.899
Swin-Transformer-T	0.935	0.964	0.945	0.971	0.858	0.921
ConvNeXt-T	0.900	0.936	0.939	0.968	0.852	0.917

TABLE I: The scores achieved by each backbone across the different test sets. As metrics are used the Intersection over Union (IoU) and Dice score.

Both models consistently provide accurate results, with SAM-HQ demonstrating improved accuracy across the different test objects.

A comparison with RITM and the baseline approach is also presented in Fig. 8. RITM often fails to accurately interpret an object's entire shape and may even merge different objects together. The advantages of SAM over RITM are also highlighted in [39]. Additionally, small errors in the projected points, for instance due to camera calibration inaccuracies, can lead to significant deviations with RITM. In contrast, both SAM and SAM-HQ effectively address these problems. Although the baseline labels offer reasonable accuracy levels, they necessitate increased annotation time and effort from the user, rendering the approach less practical.

The effect of the number of input points on accuracy is tested in Fig. 9a. In the plot, *Original* refers to the set of input points provided by the users during Sec.IV-B. Instead, *2X* and *4X* denote the conditions of sampling additional points in the middle of existing ones. Notably, the *2X* setting appears to improve accuracy, while no real benefits are observed with the *4X* case.

Ultimately, the impact of ViT model complexity on annotation accuracy is assessed in Fig. 9b. This involves evaluating both the *base* and *large* ViT models for both SAM and SAM-HQ. Additionally, SAM-HQ introduces a *tiny* variant. The figure highlights SAM-HQ's enhanced performance, even with its smallest and most resource-efficient variant.

D. Downstream Segmentation Results

To evaluate the effectiveness of the generated dataset using the proposed method, a semantic segmentation task is undertaken. The network architecture utilized is based on DeeplabV3+ [43]. In its original implementation, this model

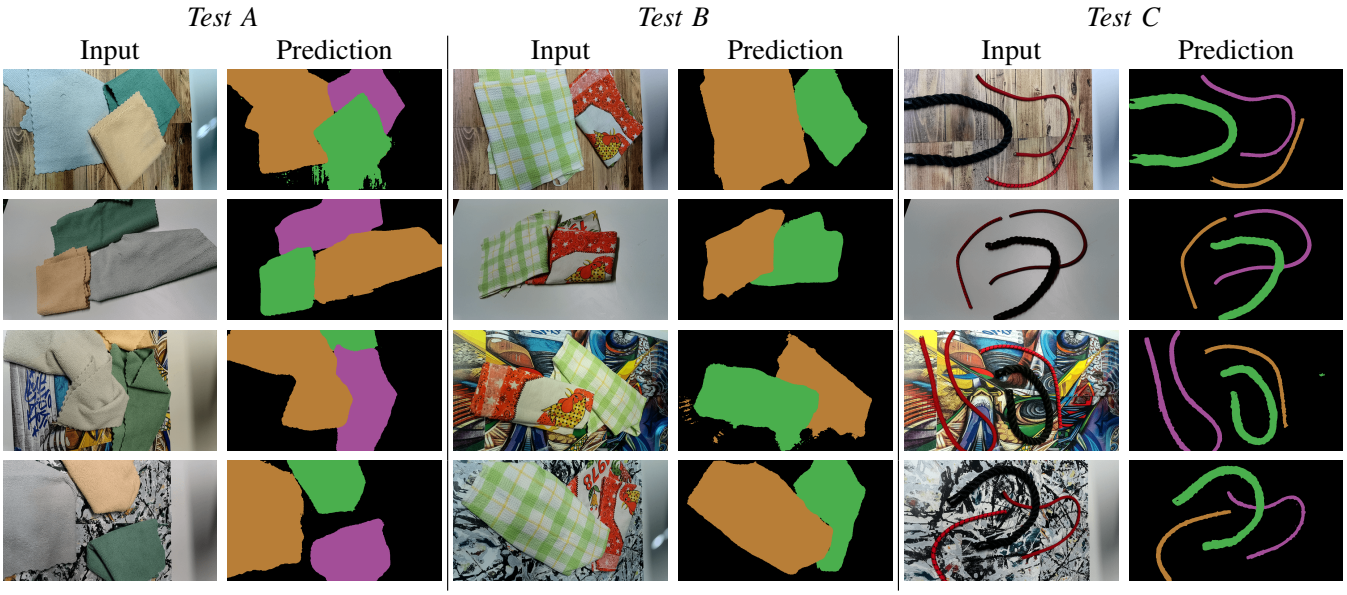


Fig. 10: Qualitative results of semantic segmentation are shown using datasets obtained via the proposed method during training. The predictions visualized are from the SwinT backbone. Each row displays a specific test scenario.

employs a modified *ResNet* [44] backbone with atrous convolutions as the encoder.

For a more comprehensive evaluation, other state-of-the-art backbone architectures are also tested, including *Swin-Transformer* [45] and *ConvNeXt* [46]. Specifically, the following backbones with similar complexity are employed: *ResNet-50*, *ResNet-101*, *Swin-Transformer-T*, and *ConvNeXt-T*. Therefore, a total of four architectures are tested.

The semantic segmentation models are trained with a common set of hyperparameters, specifically: model weights pre-trained on ImageNet, batch size 4, Adam optimizer with learning rate 1×10^{-4} and polynomial learning rate adjustment policy to a minimum of 10^{-8}, with power 0.95. A warmup procedure is executed for the first 1000 steps with an initial learning rate of 1×10^{-8}. The *ResNet* backbones are trained with an output stride of 16 and separable convolutions. The training is executed for a maximum of 200 epochs with an early stopping procedure executed to terminate the training process if the validation loss does not decrease for 5 epochs in a row. The best weights are saved according to the minimum validation loss. The models are implemented in PyTorch 2.0 and trained with an NVIDIA GeForce GTX 2080 Ti on an Intel Core i9-9900K CPU clocked at 3.60GHz.

For each training dataset group (see Sec. IV-A), the training set is derived from 90% of the original set, while the validation is done on the remaining 10%. The image labels are obtained exploiting SAM-HQ [42] due to the improved accuracy over SAM, see Sec. IV-C. The data augmentation scheme includes flipping, perspective distortions, random brightness, random contrast and finally resizing to 576×1024 pixels.

The segmentation results on the test sets can be found in Fig. 10 and Tab. I. Specifically, Fig. 10 presents a qualitative assessment of the segmentation outcomes, while Tab. I quantitatively evaluates performance using the IoU and Dice metrics. The Dice metric is defined as $\text{Dice} = 2\frac{|M_p \cap M_{gt}|}{|M_p| + |M_{gt}|}$, where M_{gt} is the ground truth and M_p is the prediction of the network. For both metrics, the average among the object classes is computed as the final score for each test sample, with 3 for *Test B* and *C* and 4 for *Test A*.

From Tab. I, it is clear that all the model architectures can segment the objects with a good level of accuracy, i.e. 90% or more for the Dice score. These results are further confirmed by close inspection of the predictions shown in Fig. 10, obtained by the *Swin-Transformer-T* backbone.

V. Limitations and Conclusions

In conclusion, this paper addresses the challenge of labeling Deformable Objects (DOs) to generate a real-world, task-specific dataset for use in data-driven methods for robotic perception. The proposed method offers an effective pixel-level labeling approach for DOs in images, utilizing sparse annotations of key points as a starting point. The utilization of the Segment Anything Model (SAM) enables us to obtain accurate dense masks without the need for specific fine-tuning objectives or domain-specific knowledge. Moreover, the proposed approach drastically improves the usability while reducing the labeling effort.

The proposed approach is subject to certain limitations. Specifically, one primary constraint lies in the simple projection step and camera setup, where objects with significant depth may introduce errors in their projection onto the same object and in cases of occlusion. To mitigate this issue, the utilization of 3D cameras is suggested, as they enable the acquisition of the individual depth values of key points and facilitate the evaluation of potential occlusions during the projection of 3D points onto the image plane.

Another limitation pertains to the restricted variability within the scene. While this method permits the labeling of

only one sample and the gathering of n data samples for training, the variability among these n samples is somewhat limited, necessitating multiple annotations and collections of diverse scenarios for comprehensive training purposes.

In future work, we will investigate implementing autonomous regeneration of scene configurations using a robotic arm to increase data variance during sample collection.

REFERENCES

[1] J. Pile, G. B. Wanna, and N. Simaan, "Force-based flexible path plans for robotic electrode insertion," in *Proc. of the ICRA*, 2014, pp. 297–303.

[2] R. J. M. Masey, J. O. Gray, T. J. Dodd, and D. G. Caldwell, "Guidelines for the design of low-cost robots for the food industry," *Industrial Robot: An International Journal*, 2010.

[3] J. Trommnau, J. Kühnle, J. Siegert, R. Inderka, and T. Bauernhansl, "Overview of the state of the art in the production process of automotive wire harnesses, current research and future trends," *Procedia CIRP*, 2019.

[4] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, p. eabd8803, 2021.

[5] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, J. Pan, W. Yuan, and M. Gienger, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.

[6] F. Zhang and Y. Demiris, "Learning garment manipulation policies toward robot-assisted dressing," *Science robotics*, 2022.

[7] 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*, 2023.

[8] H. G. Nguyen, R. Habiboglu, and J. Franke, "Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing," *Procedia CIRP*, 2022.

[9] R. Firoozi, J. Tucker, S. Tian, Majumdar *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[10] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[11] A. Caporali, K. Galassi, B. L. Žagar, R. Zanella, G. Palli, and A. C. Knoll, "RT-DLO: Real-time deformable linear objects instance segmentation," *IEEE Transactions on Industrial Informatics*, 2023.

[12] A. Caporali, K. Galassi, and G. Palli, "Deformable linear objects 3D shape estimation and tracking from multiple 2D views," *IEEE Robotics and Automation Letters*, 2023.

[13] S. Pirozzi and C. Natale, "Tactile-based manipulation of wires for switchgear assembly," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 6, pp. 2650–2661, 2018.

[14] K. P. Cop, A. Peters, B. L. Žagar, D. Hettegger, and A. C. Knoll, "New metrics for industrial depth sensors evaluation for precise robotic applications," in *IEEE/RSJ Int. Conf. IROS*. IEEE, 2021.

[15] R. Zanella, A. Caporali, K. Tadaka, D. De Gregorio, and G. Palli, "Auto-generated wires dataset for semantic segmentation with domain-independence," in *2021 International Conference on Computer, Control and Robotics (ICCCR)*. IEEE, 2021, pp. 292–298.

[16] J. Dirr, D. Gebauer, J. Yao, and R. Daub, "Automatic image generation pipeline for instance segmentation of deformable linear objects," *Sensors*, 2023.

[17] W. Wu, Y. Zhu, X. Zheng, and Y. Guo, "A novel cable-grasping planner for manipulator based on the operation surface," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102252, 2022.

[18] C. Dai, G. Shan, H. Liu, C. Ru, and Y. Sun, "Robotic manipulation of sperm as a deformable linear object," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 2799–2811, 2022.

[19] Y. Song, K. Yang, X. Jiang, and Y. Liu, "Vision based topological state recognition for deformable linear object untangling conducted in unknown background," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2019, pp. 790–795.

[20] X. Huang, D. Chen, Y. Guo, X. Jiang, and Y. Liu, "Untangling multiple deformable linear objects in unknown quantities with complex backgrounds," *IEEE Transactions on Automation Science and Engineering*, 2023.

[21] S. Jin, W. Lian, C. Wang, M. Tomizuka, and S. Schaal, "Robotic cable routing with spatial representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5687–5694, 2022.

[22] Z. Sun, H. Zhou, L. Nanbo, L. Chen, J. Zhu, and R. B. Fisher, "A robust deformable linear object perception pipeline in 3d: From segmentation to reconstruction," *IEEE Robotics and Automation Letters*, pp. 1–13, 2023.

[23] D. Zheng, S. Yao, W. Xu, and C. Lu, "Differentiable cloth parameter identification and state estimation in manipulation," *IEEE Robotics and Automation Letters*, 2024.

[24] P. Jiménez and C. Torras, "Perception of cloth in assistive robotic manipulation tasks," *Natural Computing*, vol. 19, pp. 409–431, 2020.

[25] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1703–1708.

[26] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg, "Speedfolding: Learning efficient bimanual folding of garments. in 2022 ieee," in *RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022.

[27] Z. Huang, X. Lin, and D. Held, "Self-supervised cloth reconstruction via action-conditioned cloth tracking," in *ICRA*. IEEE, 2023.

[28] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," *preprint arXiv:1412.7144*, 2014.

[29] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari, "Training object class detectors with click supervision," in *CVPR*, 2017.

[30] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *ICCV*. IEEE, 2001.

[31] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *CVPR*, 2016.

[32] H. Zhang, Y. Zeng, H. Lu, L. Zhang, J. Li, and J. Qi, "Learning to detect salient object with multi-source weak supervision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[33] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *CVPR*, 2019.

[34] B. Lutz, L. Janisch, D. Kisskalt, and D. Regulin, "Interactive Image Segmentation Using Superpixels and Deep Metric Learning for Tool Condition Monitoring," in proceeding.

[35] G. Zheng, A. H. Awadallah, and S. Dumais, "Meta label correction for learning with weak supervision," 2019.

[36] J. Liu, Q. Wang, H. Fan, S. Wang, W. Li, Y. Tang, D. Wang, M. Zhou, and L. Chen, "Automatic label correction for the accurate edge detection of overlapping cervical cells," *arXiv preprint arXiv:2010.01919*, 2020.

[37] K. Sofiiuk, I. A. Petrov, and A. Konushin, "Reviving iterative training with mask guidance for interactive segmentation," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 3141–3145.

[38] S. Ren, F. Luzi, S. Lahrichi, K. Kassaw, L. M. Collins, K. Bradbury, and J. M. Malof, "Segment anything, from space?" in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 8355–8365.

[39] M. A. Mazurowski, H. Dong, H. Gu, J. Yang, N. Konz, and Y. Zhang, "Segment anything model for medical image analysis: an experimental study," *Medical Image Analysis*, vol. 89, p. 102918, 2023.

[40] M. Pantano, V. Klass, Q. Yang, A. Sathuluri *et al.*, "Simplifying robot grasping in manufacturing with a teaching approach based on a novel user grasp metric," in *5th Int. Conf. on Industry 4.0 and Smart Manufacturing*, 2023.

[41] J. Brooke, *SUS - A quick and dirty usability scale*. CRC Press, 1996.

[42] L. Ke, M. Ye, M. Danelljan, Y.-W. Tai, C.-K. Tang, F. Yu *et al.*, "Segment anything in high quality," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[43] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE/CVF CVPR*, 2016.

[45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. of the IEEE/CVF CVPR*, 2021.

[46] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proc. of the IEEE/CVF CVPR*, 2022.