

Curve Fitting-Based Deformation Tracking for Vision-Based Robotic Applications

Abhaya Pal Singh ^{1,*}, Dmytro Romanov ¹, Ekrem Misimi ², and Alex Mason ¹

¹ Faculty of Science and Technology, Norwegian University of Life Sciences (NMBU), Ås, Norway

² Robotics and Intelligent Automation, SINTEF Ocean AS, Trondheim, Norway

Email: abhaya.aps@gmail.com (A.P.S.); dmytro.romanov@nmbu.no (D.R.); ekrem.misimi@sintef.no (E.M.); alex.mason@nmbu.no (A.M.)

*Corresponding author

Abstract—Application of robotics on production lines often involves handling flexible objects (such as items of natural origin or plastic bags containing liquid/bulk substances), which makes it crucial to consider the shape of an item before and after it has been affected by robotic manipulation. Most of the time deformable items are challenging for the robot in such operations as grasping, cutting, or packaging. The objective of this paper is to track object deformations and perform a task based on this information. The paper addresses issues in tracking object deformation and proposes a solution for deformation tracking to form preliminary knowledge and scene awareness on the robot side. A curve-fitting-based method was implemented to define a region of interest using images from a RealSense D415 camera. The developed approach identifies the maximum number of aligned points and uses it to determine where the deformation occurred. The results of this research show that the deformations are efficiently tracked. Utilising the algorithm proposed in this paper, an efficient method capable of making the robot aware of the deformation present in the scene is demonstrated. This approach is applicable in domains such as food processing, healthcare, and other fields where gentle and precise manipulations are required. The method is useful in industrial applications in which deformation cannot be completely avoided but still needs to be tracked.

Keywords—camera, curve fitting, deformation, manipulation, robot, tracking, vision, control

I. INTRODUCTION

An Interaction with deformable objects (i.e., those which change shape during manipulation) is a hot topic right now in robotics [1]. This is an enormous development because robots that can handle these objects become more flexible [2] and can be utilized in a variety of industries, including services [3], manufacturing [4], and healthcare [5]. These deformable objects are, however, significantly more challenging to manipulate than conventional solid objects [6]. Researchers are still working to address this yet-to-be-resolved challenge [7]. We need to advance in several areas of robotics,

including the design of better robot hardware [8], the development of more intelligent methods for tracking object deformations [9], the development of methods for planning actions [10], and the supervision of the robot's actions [11], to meet this challenge.

A vision-based control strategy based on the Kelvin constitutive equation and the standard spring-mass model is proposed to execute autonomous cutting tasks for deformable objects [12]. Lines and distance are used as features to define the knife position on the picture plane to track the targeted surface. The algorithm's functioning has been demonstrated by the experimental results using various materials, including sponges, artificial tissues, and pork liver. Aranda *et al.* [13] proposed a template-based shape servoing scheme to track the deformation to solve two problems, i.e. if significant deformations exist, a vision-based tracking approach with unconstrained problems was developed, and a solution for handling the underactuated shape control was proposed. Rastegarpana *et al.* [14] proposed a novel method for tracking deformation in real-time. They created an algorithm for splitting a point cloud into smaller ones, with a little variance. They applied this information to a robot that tracks deformation in the actual environment using an RGBD camera. To address the limitations of initial conditions, Keipour *et al.* [15] propose a method for detecting deformable objects that can handle crossings and occlusions and can be used for tasks such as routing and manipulation, as well as automatically providing the initialization required by tracking methods. Pecyna *et al.* [16] investigated the use of both visual and tactile inputs in completing a task of tracking deformable objects. They established a benchmark in simulation and illustrated how reinforcement learning aids in improving visual-tactile fusion compared to using single sensing inputs, and the findings demonstrated the importance of multimodality, that vision played a vital role in completing the task. Zuern *et al.* [17] proposes a method for deformable object localization and tracking. The suggested method registers a model of a de-formable object to a point cloud acquired from a stereo vision system monitoring the scene using self-organizing maps. A unique approach for deformable tracking in deformable

Manuscript received June 23, 2023; revised August 14, 2023; accepted August 28, 2023; published March 8, 2024.

SLAM is also reported in the available research [18]. Each map point is used to create a 3D surfel that is a local approximation of the scene surface. The deformations of the map are modeled by moving these surfels and proposed to remove any link between 3D map points, which has also been proven experimentally. To track the deformation of deformable objects in real-time, a Gaussian mixed model [19] with uniform noise distribution is developed for point registration, and the estimates are then sent to a dynamic simulation module for further refinement. A set of experiments validate this established concept.

According to the literature, there is no one-of-a-kind method, or a unique approach is there for tracking the deformation of a deformable object. It is because of the recent developments of different novel contributions in this particular area. The research contributed to the field of deformation tracking by introducing a novel notion based on curve fitting. The approach outlined in this research reads the ROS bag file to retrieve image and depth data recorded by a vision system (Red, Green, Blue plus Depth (RGB-D) camera). All the images are processed, the corresponding depth is read, and an “Area of Interest (AOI): is displayed. The depth values inside the AOI are evaluated to establish the maximum depth values, and the relevant index for each row is utilized to fit a curve that will track down the deformation.

The following describes the structure of the paper: A brief summary of recent events, which includes the contribution provided by this article, is included in Section I. Section II has the details of the notations that have been used in this paper, and Section III presents the problem description that has been addressed in the paper with details of the setup used in Section IV. The main results and discussion, and conclusion are presented in Sections V and VI, respectively.

II. NOTATIONS

In this section, we present the notations that have been used in this paper. Let $\mathcal{A} \in \mathbb{R}^{N \times M \times 2}$ be a matrix that has RGB, and depth information recorded from an RGB-D camera. Let $C \in \mathbb{R}^{N \times M} \subset \mathcal{A}$ be a matrix that has all the color information present in a particular scene (at each pixel) of a $N \times M$ resolutions RGB camera. Let $\mathcal{D} \in \mathbb{R}_+^{N \times M} \subset \mathcal{A}$ be a positive matrix that has all the depth information present in a particular scene (at each pixel) of a $N \times M$ resolutions depth camera. Let $d \in \mathbb{R}_+^{n \times m} \subset \mathcal{D}$ is also a positive matrix with $n \subset N$, $m \subset M$, $n \leq N$ and $m \leq M$. Let \mathcal{O} be an operator that maps $\mathcal{D} \mapsto d \Rightarrow d = \mathcal{O}(\mathcal{D})$ and captures depth information of particular pixels of interest. Let $\mathbf{M}\mathbf{X}$ be an operator that maps $d_+ \in \mathbb{R}^{n \times m} \mapsto d_{\max} \in \mathbb{R}_+^{n \times 1} \Rightarrow d_{\max} = \mathbf{M}\mathbf{X}(d)$, that gives the maximum of each row of the matrix d . The term “area of interest (AOI)” is used in this paper to indicate a pixel of interest where further analysis is required.

III. PROBLEM FORMULATION AND METHOD

When robotic manipulation is involved, tracking the deformation of a deformable object is challenging and important. The deformable object’s shape, size, and orientation may have changed as a result of the robotic manipulation, and thus the tracking provides information on the next steps in the robotic manipulation without destroying the deformable object. In this paper, the deformation to be tracked is caused by a disturbance (force applied) to a sponge (a deformable object), as shown in Fig. 1, which changes the shape of the sponge. Looking at Fig. 1(a) and Fig. 1(b), one can easily conclude that the shape of the sponge changed after applying force to it, and the upper horizontal piece of the sponge shifted downward. The goal now is to track this deformation so that the robot can be given information about this AOI and do further manipulations such as cutting at this AOI or marking at this AOI, and so on.

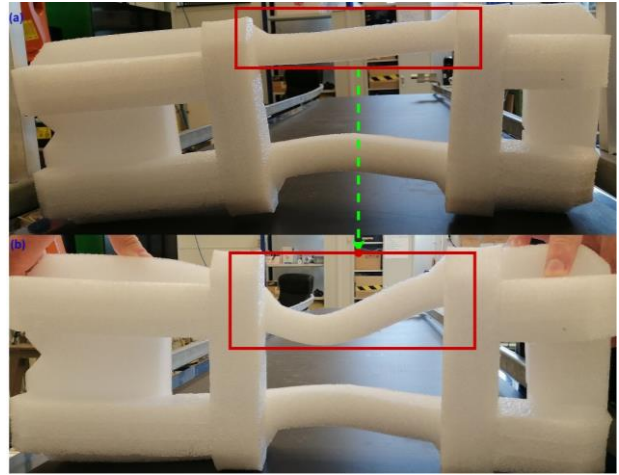


Fig. 1. The deformable object Sponge: (a) when there is no disturbance, (b) when there is a disturbance.

The method used for deformation tracking is based on the adoption of an intriguing idea focused on curve fitting. In this approach, the ROS bag file needs to be read for the purpose to access image and depth data captured by a vision system. The depth values within the AOI are evaluated to determine the maximum depth values, and the relevant indices for each row (that has the maximum values) are used to fit a curve to track the deformation of the object. All the processing is implemented using MATLAB.

IV. THE SETUP

In this paper, a real-sense D415 depth camera is installed at the position depicted in Fig. 2, and RGB-D video is recorded to capture the full information about the deformation on the sponge caused by the disturbance. This camera placement of Fig. 2 is intended to mimic the visual information that can be read from an eye-to-hand configuration on a robot, as shown in Fig. 3. The process of image and video processing is then carried out to find the deformation at the AOI after the video has been captured.

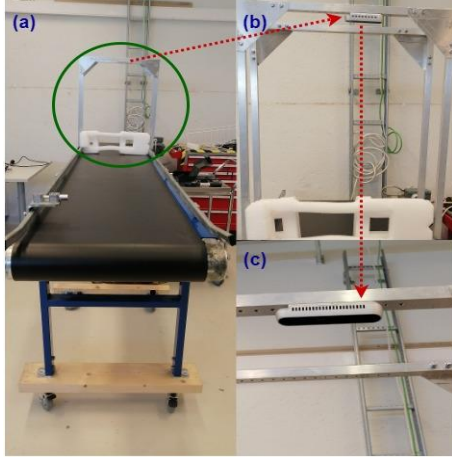


Fig. 2. Real-sense camera placement for the video recording.

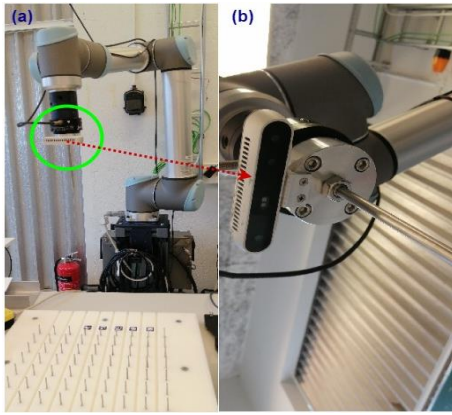


Fig. 3. Real-sense camera placement on the robot in eye-to-hand configuration.

V. MAIN RESULTS AND DISCUSSION

In this section, the steps of tracking the deformation in the deformable object are discussed step by step. The first step is to detect the deformable object itself and Algorithm 1 has been used to detect it. Let us define the total number of white objects in the scene as WO , which has information about how many white objects are present in the scene, and an operator \mathbf{WOJ} that maps $\mathcal{A} \in \mathbb{R}_+^{N \times M \times 2} \mapsto WO \in \mathbb{R}_+^{1 \times 1} \Rightarrow WO = \mathbf{WOJ}(\mathcal{A})$. The output of the Algorithm 1 can be seen in Fig. 4.

Algorithm 1: To detect the sponge in the scene.

Input: Obtain \mathcal{A}
Output: A rectangular box around $\mathbf{WOJ}(\mathcal{A})$

- 1 Step 1: Read \mathcal{A} from the bag file;
- 2 Step 2: Get the info message from \mathcal{A} ;
- 3 **for** $i = 1$: number of messages from \mathcal{A} **do**
- 4 Step 3: Marker and mask \mathcal{A} to detect white objects;
- 5 Step 4: Get the properties of the white objects;
- 6 **for** $j = 1$: $length(WO)$ **do**
- 7 Step 5: Read the area of the WO 's;
- 8 **if** $area(WO) > specific\ area$ **then**
- 9 Step 6: Display a rectangle at the AOI;
- 10 **end**
- 11 **end**
- 12 Step 7: Display the detected white object in the scene;
- 13 **end**

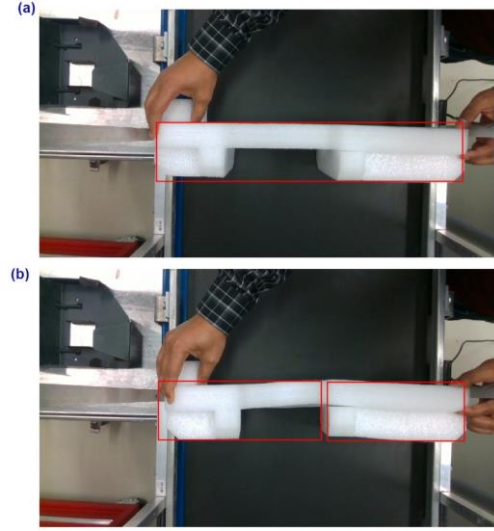


Fig. 4. The detected sponge in the scene, (a) when there is no disturbance, and (b) when there is a disturbance.

From Fig. 4, it can be seen that the object is detected in the scene. Now, let us modify the algorithm to find the locations of all the points based on the depth information \mathcal{D} present at the detected image of Algorithm 1, and display the places where the maximum depth is present at the AOI. At this stage, we now have the information of $C \in \mathbb{R}^{N \times M} \subset \mathcal{A}$, the color information present in a particular scene (at each pixel) of a $N \times M$ resolutions RGB camera, and $\mathcal{D} \in \mathbb{R}_+^{N \times M} \subset \mathcal{A}$, the depth information present in a particular scene (at each pixel) of a $N \times M$ resolutions depth camera.

The modified Algorithm 2 will give the output as shown in Fig. 5.

Algorithm 2: To detect all the maximum points at the AOI in the scene.

Input: Obtain \mathcal{A}
Output: A rectangular box around $\mathbf{WOJ}(\mathcal{A}) \cup (\mathcal{O}(\mathcal{D}) \cap \mathbf{MX}(d))$

- 1 Step 1: Read \mathcal{A} from the bag file;
- 2 Step 2: Get C and \mathcal{D} from \mathcal{A} ;
- 3 **for** $i = 1$: number of messages from \mathcal{A} **do**
- 4 Step 3: Marker and mask \mathcal{A} to detect white objects;
- 5 Step 4: Get $\mathcal{O}(\mathcal{D})$;
- 6 Step 5: $[Value, Index] = \mathbf{MX}(d)$
- 7 **for** $j = 1$: $length(Index)$ **do**
- 8 Step 6: Insert a point at $\mathbf{MX}(d)$
- 9 **end**
- 10 Step 7: Display the scene;
- 11 **end**

It is clear from Fig. 5 that the objective of collecting all the maximum points is fulfilled. Now to track the deformation where the maximum depth is aligned, it is evident to fit a straight line (assuming the deformation is represented by a line and is obvious) at the point where $\mathbf{MX}(d)$ is positioned into the matrix d . This straight line can be fetched using the curve fitting concept at the locations detected by $\mathbf{MX}(d)$ at the AOI. Let us further

modify the algorithm to fit a line at the locations of the d_{\max} in d.

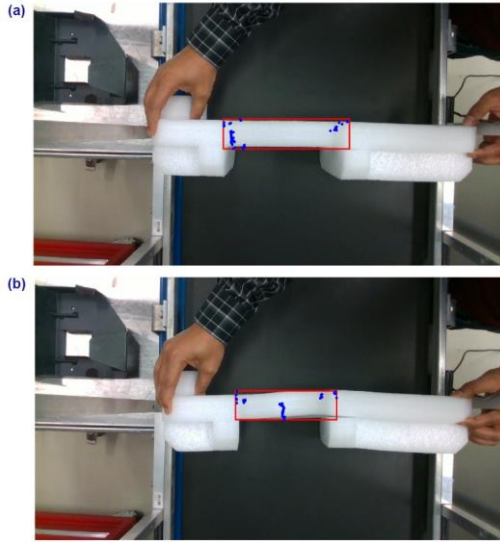


Fig. 5. Detected AOI with $\mathbf{MX}(d)$ information, (a) when there is no disturbance, and (b) when there is a disturbance.

In Algorithm 3, the terms *numel*, *unique* are standard MATLAB commands. The output of this algorithm is shown in Fig. 6. It is clear from this figure that the deformation is tracked and displayed in the scene (Fig. 6) in green color.

Algorithm 3: To detect all the maximum points at the AOI in the scene and fit a line at the deformation.

```

Input: Obtain  $\mathcal{A}$ 
Output: A rectangular box around  $\mathbf{WOI}(\mathcal{A}) \cup (\mathcal{O}(\mathcal{D}) \cap \mathbf{MX}(d))$  and a fitted line at  $d_{\max}$  in d
1 Step 1: Read  $\mathcal{A}$  from the bag file;
2 Step 2: Get  $C$  and  $\mathcal{D}$  from  $\mathcal{A}$ ;
3 for  $i = 1$ : number of messages from  $\mathcal{A}$  do
4     Step 3: Marker and mask  $\mathcal{A}$  to detect white objects;
5     Step 4: Get  $\mathcal{O}(\mathcal{D})$ ;
6     Step 5:  $[Value, Index] = \mathbf{MX}(d)$ 
7     for  $j = 1$ : length(Index) do
8         Step 6: Insert a point at  $\mathbf{MX}(d)$ 
9     end
10    Step 7: Get unique(Index);
11    for  $k = 1$ : numel(unique(Index)) do
12        Step 8: Check if the maximum of points is aligned along the vertical axis;
13        Step 9: Find the horizontal coordinate with the maximum number of aligned points;
14    end
15    Step 10: Fit line on the image;
16    Step 11: Display the scene;
17 end
    
```

The final Algorithm 3 reads the bag file containing the image and depth data. Selects specific image and camera information topics from the bag file and stores in $\mathcal{A} \in \mathbb{R}^{N \times M \times 2}$. The frame labels for object detection are obtained using camera information. All the images are processed,

the corresponding depth $\mathcal{D} \in \mathbb{R}_+^{N \times M}$ and images $C \in \mathbb{R}^{N \times M}$ are read, and an AOI is displayed. The depth values d within the AOI are analyzed to determine the maximum depth value $\mathbf{MX}(d)$ and its corresponding index for each row. The image is annotated with blue points that represent the detected $\mathbf{MX}(d)$ and is displayed around the AOI. The algorithm determines whether there are any aligned points along the vertical axis within a given bandwidth. The algorithm finds the horizontal coordinate with the greatest number of aligned points and draws a vertical line on the image in green color to display (track down) the deformation. The reliability and correctness of the algorithm can also be verified by the 3D visualization generated from the real-sense viewer software, as shown in Fig. 7. In this figure, when no force is applied, i.e., Fig. 7(a), the deformation tracking based on curve fitting is aligned towards the left-hand side of the AOI (see Fig. 6(a)), simply because the depth reading from the camera is maximum at the left-hand side, as shown in this figure. When a force is applied, as in Fig. 7(b), the deformation tracking based on curve fitting is aligned around the middle of the AOI (see Fig. 6(b)), and this information can be easily concluded when referencing Fig. 7(b), which tells us that there is a maximum depth around the middle of the AOI.

It is obvious that if the deformation tracking is successful, the robot (or operator) must be made aware of these details. The robotic arm then targets this location and performs the necessary marking or cutting tasks as instructed. Further altering the algorithm, the equation of the line representing the aligned points as well as the depth value of those points can be achieved. The outcomes can be displayed and saved for later analysis. The algorithm proposed is useful in domains such as food processing, healthcare, and other fields where gentle and precise manipulations are required. The approach is beneficial in industrial settings where deformation cannot be prevented fully but must still be supervised based on the information obtained for further manipulations.

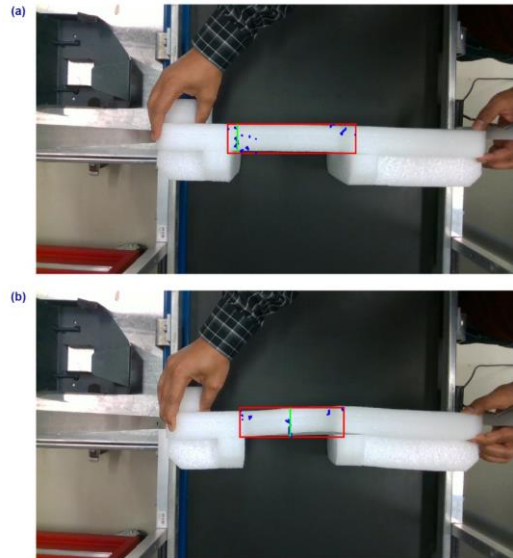


Fig. 6. Tracking of the deformation through the curve fitting, (a) when there is no disturbance, and (b) when there is a disturbance.



Fig. 7. 3D visual of the scene, (a) when there is no disturbance, and (b) when there is a disturbance.

VI. CONCLUSION

In this paper, the tracking of deformation has been carried out satisfactorily. A curve-fitting approach is used for deformation tracking, and a line is fitted at the point of interest. A real-sense D415 depth camera is attached so that it mimics the eye-in-hand configuration on a robot, and RGB-D data is collected to provide a comprehensive understanding of the deformation on the sponge induced by a disturbance. Algorithms are described in the study to achieve the motivating goal of tracking the deformation. A demonstration video showcasing the proposed approach is available at <https://youtu.be/HZ6BcyDFyuc>.

The suggested approach has limitations in that it currently fits a linear straight line at the obtained maximum values of the points read by the depth camera while avoiding outliers in a targeted AOI. Future work is aimed at experimentally validating the results obtained on a real robot that can be guided to move to the location of the AOI as suggested by the algorithm, and perform the tasks as instructed.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

A. P. S.: Conceptualization of this study, investigation, methodology, writing—original draft preparation, writing—review and editing. D. R.: Writing—review and

editing. E. M.: Conceptualization of this study, supervision, project administration, funding acquisition. A. M.: Conceptualization of this study, supervision, writing—review, and editing, project administration, funding acquisition; all authors had approved the final version.

FUNDING

This work is supported by the Norwegian Research Council project “GentleMAN-Gentle and Advanced Robotic Manipulation of 3D Compliant Objects”, project no. 299757. It is also supported, in part, by the European Commission project “RoBUTCHER-A Robust, Flexible and Scalable Cognitive Robotics Platform”, grant agreement id 871631”.

REFERENCES

- [1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, A. Farshid, D. Berenson, F. Ficuciello *et al.* “Challenges and outlook in robotic manipulation of deformable objects,” *IEEE Robotics & Automation Magazine*, vol. 29, no. 3, pp. 67–77, 2022.
- [2] B. Jean-François, J. Lintuvuori, C. Lacouture, T. Barois, A. Deblais, K. Xie, S. Cassagnere *et al.* “From collections of independent, mindless robots to flexible, mobile, and directional superstructures,” *Science Robotics*, vol. 6, no. 56, eabd0272, 2021.
- [3] M. Luz, J. Ruiz-del-Solar, S. Li, J. P. Siebert, and G. Aragon-Camarasa, “Continuous perception for deformable objects understanding,” *Robotics and Autonomous Systems*, vol. 118, pp. 220–230, 2019.
- [4] D. Andronas, Z. Arkouli, N. Zacharaki, G. Michalos, A. Sardelis, G. Papanikolopoulos, and S. Makris, “On the perception and handling of deformable objects—A robotic cell for white goods industry,” *Robotics and Computer-Integrated Manufacturing*, vol. 77, 102358, 2022.
- [5] Y. Zhang and L. Mingyue, “A review of recent advancements in soft and flexible robots for medical applications,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 16, no. 3, e2096, 2020.
- [6] S. Cui, R. Wang, J. Wei, F. Li, and S. Wang, “Grasp state assessment of deformable objects using visual-tactile fusion perception,” in *Proc. 2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 538–544.
- [7] E. Veronica, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, and J. L. Wyatt, “Modeling of deformable objects for robotic manipulation: A tutorial and review,” *Frontiers in Robotics and AI*, 2020, vol. 82.
- [8] Z. Xu, C. Cheng, B. Burchfiel, E. Cousineau, S. Feng, and S. Song, “Dexterity: Deformable manipulation can be a breeze,” *arXiv preprint arXiv:2203.01197*, 2022.
- [9] M. Yu, L. Kangchen, H. Zhong, S. Song, and X. Li, “Global model learning for large deformation control of elastic deformable linear objects: An efficient and adaptive approach,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 417–436, 2022.
- [10] W. Yan, A. Vangipuram, P. Abbeel, and L. Pinto, “Learning predictive representations for deformable objects using contrastive estimation,” in *Proc. Conference on Robot Learning*, 2021, pp. 564–574.
- [11] W. Benjamin, A. Cosgun, R. Newbury, T. Tran, W. P. Chan, T. Drummond, and E. Croft, “Demonstrating cloth folding to robots: Design and evaluation of a 2d and a 3d user interface,” in *Proc. 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, IEEE, 2021, pp. 155–160.
- [12] H. Lijun, H. Wang, Z. Liu, W. Chen, and X. Zhang, “Vision-based cutting of deformable objects with surface tracking,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 4, pp. 2016–2026, 2020.
- [13] A. Miguel, J. A. C. Ramon, Y. Mezouar, A. Bartoli, and E. Özgür, “Monocular visual shape tracking and servoing for isometrically deforming objects,” in *Proc. 2020 IEEE/RSJ International*

- Conference on Intelligent Robots and Systems (IROS)*, pp. 7542–7549, 2020.
- [14] R. Alireza, R. Howard, and R. Stolkin, “Tracking linear deformable objects using slicing method,” *Robotica*, vol. 40, no. 4, pp. 1188–1206, 2022.
- [15] K. Azarakhsh, M. Bandari, and S. Schaal, “Deformable one-dimensional object detection for routing and manipulation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4329–4336, 2022.
- [16] P. Leszek, S. Dong, and S. Luo, “Visual-tactile multimodality for following deformable linear objects using reinforcement learning,” in *Proc. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3987–3994.
- [17] Z. Manuel, M. Wnuk, A. Schneider, A. Lechler, and A. Verl, “Localization and tracking of deformable linear objects with self organizing maps,” in *Proc. ISR Europe 2022; 54th International Symposium on Robotics*, pp. 1–9. VDE, 2022.
- [18] L. Jose, J. J. G. Rodríguez, J. D. Tardos, and J. M. M. Montiel, “Direct and sparse deformable tracking,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11450–11457, 2022.
- [19] T. Te and M. Tomizuka, “Track deformable objects from point clouds with structure preserved registration,” *The International Journal of Robotics Research*, vol. 41, no. 6, pp. 599–614, 2022.

Copyright © 2024 by the authors. This is an open access article distributed under the Creative Commons Attribution License ([CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/)), which permits use, distribution and reproduction in any medium, provided that the article is properly cited, the use is non-commercial and no modifications or adaptations are made.