Learning to Estimate 3-D States of Deformable Linear Objects from Single-Frame Occluded Point Clouds

Kangchen Lv, Mingrui Yu, Yifan Pu, Xin Jiang, Gao Huang, and Xiang Li

***Abstract*— Accurately and robustly estimating the state of deformable linear objects (DLOs) is crucial for DLO manipulation and other applications. This paper focuses on learning to robustly estimate the states of DLOs from single-frame point clouds in the presence of occlusions using a data-driven method. We propose a novel two-branch network architecture to exploit global and local information of input point cloud respectively and design a fusion module to effectively leverage the advantages of both branches. Simulation and real-world experimental results demonstrate that our method can generate globally smooth and locally precise DLO state estimation results even with heavily occluded point clouds.**

I. Introduction

Robotic manipulation of deformable linear objects (DLOs), such as ropes and wires, has a wide variety of applications [1], [2]. However, the infinite dimensional state space, frequent occlusions and noises make it very challenging to estimate the DLO state accurately. Commonly used DLO state representations include Fourier-based parameterization [3], implicit latent descriptors [4], [5], a chain of nodes [6]–[9], etc. Among these methods, representing a DLO as a chain of uniform 3-D nodes (see Fig. 1) is general in various manipulation tasks and will be adopted in this work.

A complete processing stream to estimate the DLO state can be roughly divided into three procedures: segmentation, detection, and tracking. With raw RGB input, some works [10]–[13] focus on how to obtain pixel-level DLO masks using traditional image processing or data-driven methods. As for detection, this step aims at estimating the positions of nodes along the DLO in one frame with the cleaned sensory data as input [14]–[16]. As for tracking, various works have also been proposed to track the correspondence of point cloud across video frames in the presence of occlusions and self-intersections [17]–[23]. However, these pure tracking-based methods rely on an accurate initial state which requires manual setting or specific initial conditions. Besides, there are few effective ways to rectify the accumulated drift errors or re-initialize for tracking failure.

In this paper, we focus on estimating a sequence of ordered and uniformly distributed nodes from single-frame point cloud occlusion-robustly to represent the state of DLO, as shown in Fig. 1. The challenges of this task are as follows: 1) there are few distinguishable features in the point cloud of DLOs; 2) occlusions and noises are common in the environment; 3) generalization ability for different DLOs is required. To deal with challenges above, we propose a novel two-branch network architecture to leverage both the global geometry information for guaranteeing smooth and occlusion-robust shape, and local geometry information for precise estimations. To the best of our knowledge, we are the first to realize accurate and robust 3-D state estimation of DLOs from single-frame point cloud input even with heavy occlusions. The whole framework is trained on synthetic dataset generated in simulation without collecting real-world data. Experiments suggest our method achieves high performance on occlusion-robust state estimation of DLOs and can be directly applied in real-world scenarios.

K. Lv, M. Yu, Y. Pu, G. Huang, and X. Li are with the Department of Automation, Tsinghua University, China. X. Jiang is with the Beijing Academy of Artificial Intelligence, Beijing, China. Corresponding author: Xiang Li (xiangli@tsinghua.edu.cn)

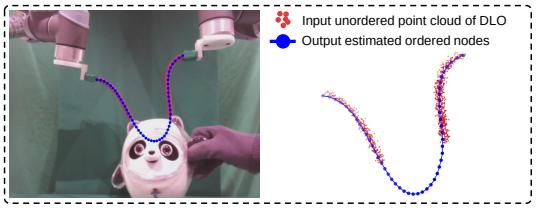


Fig. 1. Illustration of our task: 3-D occlusion-robust DLO state estimation from a single-frame point cloud. Red points are the unordered incomplete point cloud of the occluded rope and blue connected dots represent our estimated ordered node sequence as its current state.

II. Method

We represent the DLO state as a sequence of M *nodes* uniformly distributed and the problem is to estimate the coordinates of the nodes $\boldsymbol{Y} \in \mathbb{R}^{M\times 3}$ from the input point cloud $\boldsymbol{X} \in \mathbb{R}^{N\times 3}$. As shown in Fig. 2, our proposed method contains two branches: an *End-to-End Regression* branch and a *Point-to-Point Voting* branch, which focuses on the global and the local geometry information, respectively. Then, a deformable registration module is designed to leverage the advantages of both branches and output the final estimations.

A. End-to-End Regression

The most straightforward approach is to train an end-to-end network with the point cloud X as input and the node sequence Y as output. We exploit a PointNet++ [24] encoder denoted as $\boldsymbol{F}(\cdot)$ to extract deep latent features $\boldsymbol{F}(\boldsymbol{X}) \in \mathbb{R}^{N\times C_{out}}$. A max pooling layer is then applied to get the global feature which is irrelevant to the input point order and a fully-connected layer $\boldsymbol{FC}_1$ finally predicts the node sequence. The whole regression network is defined as $\boldsymbol{Y}_{\text{reg}}^{\text{pred}} = \boldsymbol{FC}_1(\text{MaxPool}(\boldsymbol{F}(\boldsymbol{X})))$.

With the ground-truth node coordinates $\boldsymbol{Y}^{\text{gt}}$, the training loss function for each sample is

$$L_{\text{reg}} = \|\boldsymbol{Y}_{\text{reg}}^{\text{pred}} - \boldsymbol{Y}^{\text{gt}}\|^2. \tag{1}$$

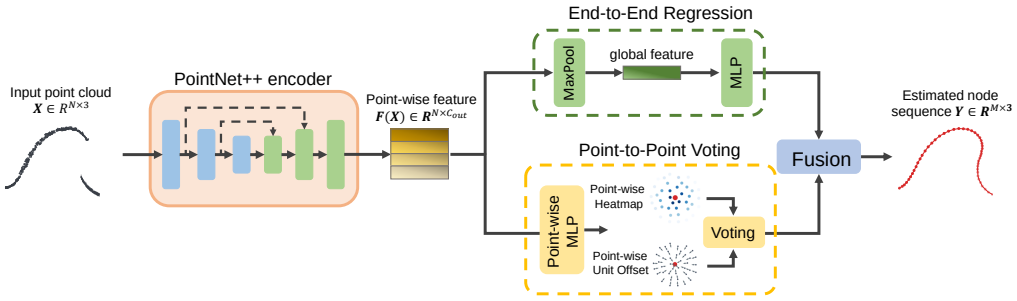


Fig. 2. Overview of the proposed method for occlusion-robustly estimating the 3-D states of DLOs. The input point cloud which might be fragmented due to occlusions is first fed into a PointNet++ encoder and the extracted features are then processed by two parallel branches: *End-to-End Regression* and *Point-to-Point Voting*. The estimation results of these two branches are finally fused with a fusion module to obtain the final output node sequence.

It is experimentally found that such a network can ensure that the estimated DLO shapes are smooth even with heavily occluded point cloud input. However, the predictions are often slightly different from the actual states such that they are not sufficiently accurate for applications (see Fig. 5).

B. Point-to-Point Voting

To make up for the shortcomings of the end-to-end regression method, we design a point-to-point voting framework to utilize local geometry information, which is inspired by early works [25], [26]. This method first defines a heatmap value H_{ij} for the distance from $\boldsymbol{x}_i$ to $\boldsymbol{y}_j$ and a unit offset vector $\boldsymbol{U}_{ij}$ for the direction. Given a neighborhood radius r, the ground-truth heatmap value H_{ij}^{gt} is defined as

$$H_{ij}^{\mathrm{gt}} = \begin{cases} 1 - \|\boldsymbol{x}_i - \boldsymbol{y}_j\|/r & , \ \|\boldsymbol{x}_i - \boldsymbol{y}_j\| < r, \\ 0 & , \ \|\boldsymbol{x}_i - \boldsymbol{y}_j\| \geq r, \end{cases} \tag{2}$$

and the ground-truth unit offset vector $\boldsymbol{U}_{ij}^{\mathrm{gt}}$ is defined as

$$\boldsymbol{U}_{ij}^{\mathrm{gt}} = \begin{cases} (\boldsymbol{y}_j - \boldsymbol{x}_i)/\|\boldsymbol{x}_i - \boldsymbol{y}_j\| & , \ \|\boldsymbol{x}_i - \boldsymbol{y}_j\| < r, \\ 0 & , \ \|\boldsymbol{x}_i - \boldsymbol{y}_j\| \geq r, \end{cases} \tag{3}$$

We then regress the point-wise heatmap $\boldsymbol{H}^{\mathrm{pred}}$ and offset vector $\boldsymbol{U}^{\mathrm{pred}}$ from the feature $\boldsymbol{F}(\boldsymbol{X})$ using point-wise fully-connected layers. The training loss for the point-to-point voting method is

$$L_{\mathrm{vot}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[(H_{ij}^{\mathrm{pred}} - H_{ij}^{\mathrm{gt}})^2 + \|\boldsymbol{U}_{ij}^{\mathrm{pred}} - \boldsymbol{U}_{ij}^{\mathrm{gt}}\|^2\right]. \tag{4}$$

The overall training loss for the whole network can be formulated as: $L_{\mathrm{tot}} = L_{\mathrm{reg}} + \alpha\, L_{\mathrm{vot}}$, where α is a pre-defined coefficient. As for the inference, the point-wise estimation for node $\boldsymbol{y}_j$ from input point $\boldsymbol{x}_i$ is obtained as $\boldsymbol{y}_j^{\mathrm{pred},i} = r\,(1 - H_{ij}^{\mathrm{pred}})\,\boldsymbol{U}_{ij}^{\mathrm{pred}} + \boldsymbol{x}_i$.

We also use the heatmap value H_{ij}^{pred} as the confidence of the prediction $\boldsymbol{y}_j^{\mathrm{pred},i}$ and only select input points with the highest K headmap value for the j^{th} node to calculate the final estimation as

$$\boldsymbol{y}_j^{\mathrm{pred}} = \left(\sum_{i\in\mathcal{K}} H_{ij}^{\mathrm{pred}}\boldsymbol{y}_j^{\mathrm{pred},i}\right) / \sum_{i\in\mathcal{K}} H_{ij}^{\mathrm{pred}}, \tag{5}$$

where the indexes of the K chosen points form the set $\mathcal{K}$.

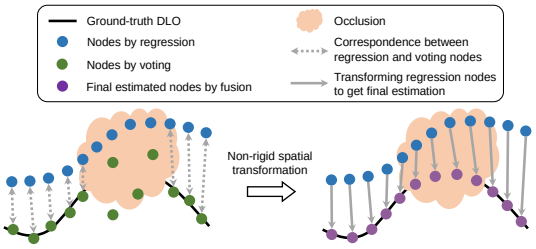


Fig. 3. Illustration of the fusion module. We first identify and exclude the occluded parts estimated by voting. Then, a non-rigid spatial transformation is estimated and the total sequence estimated by regression is transformed to get the final node sequence (purple points) using this transformation.

Experiment results show that the point-to-point voting scheme can produce precise state estimations. However, if there are no enough input points in the local neighborhood because of occlusions, the prediction of the occluded part will be significantly inaccurate (also shown in Fig. 5).

C. Fusion of the Two Branches

To leverage the advantages of both two branches and achieve occlusion-robust state estimations, we further introduce a non-rigid registration-based fusion module. We aim to estimate a non-rigid transformation from the smooth but imprecise regression results to the accurate unoccluded voting results, as shown in Fig. 3. Details of each step are described as follows:

1) Select the unoccluded node subset: Firstly, we define the visible possibility $p_j = \max_i H_{ij}^{\mathrm{pred}}$, and regard the nodes whose p_j is greater than a pre-defined threshold $T \in [0, 1]$ as unoccluded parts. We denote the subsets of unoccluded regression and voting results for the following non-rigid registration (simplified as *nrr*) as $\boldsymbol{Y}_{\mathrm{reg}}^{\mathrm{nrr}}$ and $\boldsymbol{Y}_{\mathrm{vot}}^{\mathrm{nrr}}$.

2) Estimate the non-rigid transformation: We utilize a modified *Coherent Point Drift* (CPD) algorithm [27] to estimate the non-rigid transformation with known correspondences. The classical CPD formulates registration as a GMM problem and ensures the coherent motion by representing the non-linear spatial transformation as $\mathcal{T}(\boldsymbol{Y}_{\mathrm{reg}}^{\mathrm{nrr}}) = \boldsymbol{Y}_{\mathrm{reg}}^{\mathrm{nrr}} + \boldsymbol{G}(\boldsymbol{Y}_{\mathrm{reg}}^{\mathrm{nrr}})\,\boldsymbol{W}$, where $\boldsymbol{G}(\cdot)\boldsymbol{W}$ represents the displacement function as a Gaussian Radius Basis Function Network.

For us, the correspondence of our source $Y_{\mathrm{reg}}^{\mathrm{nrr}}$ and target $Y_{\mathrm{vot}}^{\mathrm{nrr}}$ has been given by the order of nodes in the sequence. Thus, there is no need to execute the E-step and we can

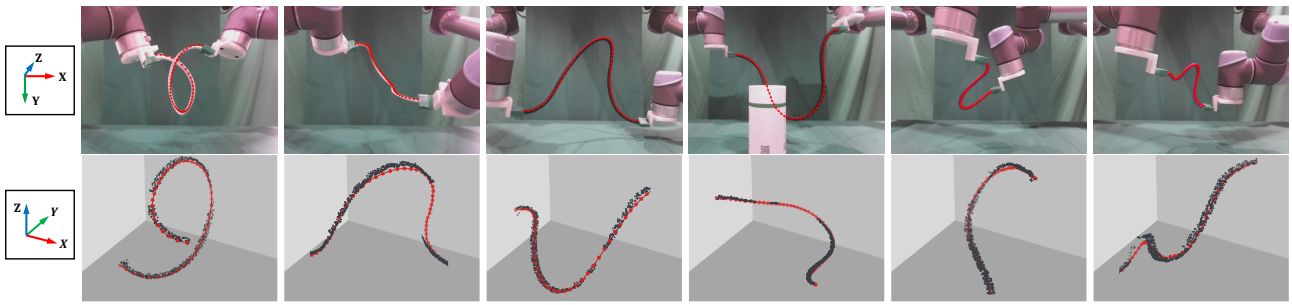
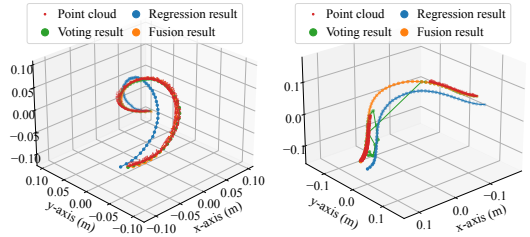


Fig. 4. State estimation results of three different real-world DLOs from occluded and fragmentary point clouds. The top row shows the raw RGB images and the reprojection of the estimated node sequence (in red color); and the bottom row shows the raw point clouds of the DLOs (in black color) and the 3-D positions of the estimated node sequence (in red color). Each column refers to a shape, in which the top image is in the common camera frame and the bottom point cloud is in a top-view frame to better illustrate the Z-axis shape.



(a) (b)

Fig. 5. Visualization of the regression, voting and fusion results of (a) unoccluded point cloud and (b) occluded point cloud.

directly update σ^2 and $\boldsymbol{W}$ iteratively. Following Eq. (22) and Eq. (23) in [27], we can fix the correspondence probability matrix as an identity matrix and solve $\boldsymbol{W}$ and σ^2 using

$$(\boldsymbol{G}(\boldsymbol{Y}_{\text{reg}}^{\text{nrr}}) + \lambda\sigma^2\boldsymbol{I})\boldsymbol{W} = \boldsymbol{Y}_{\text{vot}}^{\text{nrr}} - \boldsymbol{Y}_{\text{reg}}^{\text{nrr}}, \tag{6}$$

$$\begin{aligned}\sigma^2 =& \frac{1}{D}(\text{tr}((\boldsymbol{Y}_{\text{vot}}^{\text{nrr}})^{\mathsf{T}}\boldsymbol{Y}_{\text{vot}}^{\text{nrr}}) - 2\text{tr}((\boldsymbol{Y}_{\text{vot}}^{\text{nrr}})^{\mathsf{T}}\mathcal{T}(\boldsymbol{Y}_{\text{reg}}^{\text{nrr}})) \\ &+ \text{tr}(\mathcal{T}(\boldsymbol{Y}_{\text{reg}}^{\text{nrr}})^{\mathsf{T}}\mathcal{T}(\boldsymbol{Y}_{\text{reg}}^{\text{nrr}}))).\end{aligned} \tag{7}$$

3) Transform the whole set of regression results: Finally, we apply the estimated non-rigid transformation to the whole regression node sequence, where the displacement function is given by the Gaussian Radius Basis Function Network and previously optimized weights $\boldsymbol{W}$. The transformed regression node sequence are our final fused state estimations:

$$\boldsymbol{Y}_{\text{fus}}^{\text{pred}} = \mathcal{T}(\boldsymbol{Y}_{\text{reg}}^{\text{pred}}) = \boldsymbol{Y}_{\text{reg}}^{\text{pred}} + \boldsymbol{G}(\boldsymbol{Y}_{\text{reg}}^{\text{pred}}, \boldsymbol{Y}_{\text{reg}}^{\text{nrr}})\,\boldsymbol{W}. \tag{8}$$

III. Results

A. Simulation Results

All the training data are generated in simulations for the convenience of getting the ground-truth node positions and our model can be trained on the synthetic data in an end-to-end manner. Two visualized examples are shown in Fig. 5. It can be seen that the regression results are robust against occlusion but imprecise, while the voting results are accurate outside occlusion but extremely unreliable for occluded parts. However, our fusion method can accurately and robustly estimate the DLO state in both unoccluded and occluded scenarios.

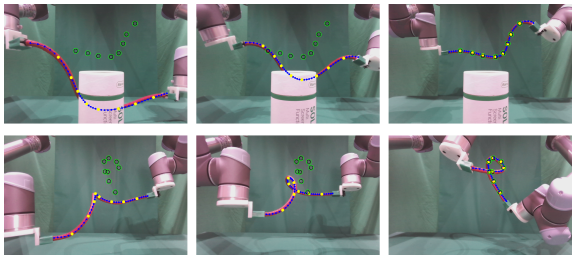


Fig. 6. Applications in downstream DLO shape control task. The blue, yellow, and green points represent the reprojected 3-D node positions, selected control nodes and target positions, respectively.

B. Real-World Experiments

We choose three DLOs of different lengths and different materials to examine the generalization ability of our method in real-world applications. The two ends of DLOs are rigidly grasped by dual robot arms and deformed to various complex shapes. Results in Fig. 4 illustrates that our method can be directly applied to estimate the real-world DLO state with small sim-to-real gaps. Even in some cases with self-intersection or occlusion by obstacles, our state estimations are still smooth and precise enough.

We also integrate our method into the DLO shape control task as the front-end perception module. As shown in Fig. 6, a uniformly-distributed subset of estimated nodes (8 yellow points) is chosen to be controlled to achieve the target positions (corresponding green points). The controller is based on our previous work [6] and our method can achieve real-time performance on a GeForce RTX 2060 GPU. In the presence of occlusions or self-intersections at both initial and middle stage of the manipulation process, our method can steadily output precise DLO states and finally achieve the target positions, which cannot be realized by the existing pure-tracking methods.

IV. Conclusions

In this work, we propose a learning-based method to robustly estimate the 3-D states of DLOs from single-frame point clouds even with heavy occlusions. We design a two-branch architecture to utilize the global or local geometry information respectively and fuse them to get the final output. The simulation and real-world experimental results demonstrate the effectiveness of our method.

REFERENCES

[1] J. Zhu, A. Cherubini, C. Dune, D. Navarro-Alarcon, F. Alambeigi, D. Berenson, F. Ficuciello, K. Harada, J. Kober, X. Li, *et al.*, "Challenges and outlook in robotic manipulation of deformable objects," *IEEE Robotics and Automation Magazine*, 2021.

[2] 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.

[3] D. Navarro-Alarcon and Y.-H. Liu, "Fourier-based shape servoing: A new feedback method to actively deform soft objects into desired 2-d image contours," *IEEE Transactions on Robotics*, vol. 34, no. 1, pp. 272–279, 2017.

[4] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, "Learning rope manipulation policies using dense object descriptors trained on synthetic depth data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9411–9418.

[5] P. Zhou, J. Zhu, S. Huo, and D. Navarro-Alarcon, "Lasesom: A latent and semantic representation framework for soft object manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5381–5388, 2021.

[6] M. Yu, K. Lv, 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*, 2022.

[7] S. Jin, C. Wang, and M. Tomizuka, "Robust deformation model approximation for robotic cable manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6586–6593.

[8] M. Yu, H. Zhong, and X. Li, "Shape control of deformable linear objects with offline and online learning of local linear deformation models," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1337–1343.

[9] M. Yu, K. Lv, C. Wang, M. Tomizuka, and X. Li, "A coarse-to-fine framework for dual-arm manipulation of deformable linear objects with whole-body obstacle avoidance," *arXiv preprint arXiv:2209.11145*, 2022.

[10] H. Dinkel, J. Xiang, H. Zhao, B. Coltin, T. Smith, and T. Bretl, "Wire point cloud instance segmentation from rgbd imagery with mask r-cnn."

[11] A. Caporali, K. Galassi, R. Zanella, and G. Palli, "Fastdlo: Fast deformable linear objects instance segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9075–9082, 2022.

[12] D. D. Gregorio, G. Palli, and L. D. Stefano, "Let's take a walk on superpixels graphs: Deformable linear objects segmentation and model estimation," in *Asian Conference on Computer Vision*. Springer, 2018, pp. 662–677.

[13] 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.

[14] S. Huo, A. Duan, C. Li, P. Zhou, W. Ma, H. Wang, and D. Navarro-Alarcon, "Keypoint-based planar bimanual shaping of deformable linear objects under environmental constraints with hierarchical action framework," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5222–5229, 2022.

[15] M. Yan, Y. Zhu, N. Jin, and J. Bohg, "Self-supervised learning of state estimation for manipulating deformable linear objects," *IEEE robotics and automation letters*, vol. 5, no. 2, pp. 2372–2379, 2020.

[16] M. Wnuk, C. Hinze, A. Lechler, and A. Verl, "Kinematic multibody model generation of deformable linear objects from point clouds," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9545–9552.

[17] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1130–1137.

[18] T. Tang, Y. Fan, H.-C. Lin, and M. Tomizuka, "State estimation for deformable objects by point registration and dynamic simulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2427–2433.

[19] T. Tang and M. Tomizuka, "Track deformable objects from point clouds with structure preserved registration," *The International Journal of Robotics Research*, p. 0278364919841431, 2018.

[20] T. Tang, C. Wang, and M. Tomizuka, "A framework for manipulating deformable linear objects by coherent point drift," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3426–3433, 2018.

[21] C. Chi and D. Berenson, "Occlusion-robust deformable object tracking without physics simulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6443–6450.

[22] Y. Wang, D. McConachie, and D. Berenson, "Tracking partially-occluded deformable objects while enforcing geometric constraints," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 199–14 205.

[23] 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.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[25] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3d hand pose estimation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 475–491.

[26] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Dense 3d regression for hand pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5147–5156.

[27] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010.