Monocular Shape Sensing for *Continuum Robot*

Thomas EnXu Li*, Jimmy Chengnan Shentu*, and Vicky Chaojun Chen*

Abstract—*Continuum Robots* have flexible structures that allow them to access confined spaces and complex environments. To control continuum robots, accurate and real-time shape sensing is essential. Currently, the visual-based method has the most potential for 3D shape estimation given its ability to provide accurate results; however, the majority of the visual-based approaches for *continuum robots* are not suitable in real-life scenarios due to their dependency on markers or multiple cameras. To address this, we propose a robust and efficient monocular-based shape sensing model that does not rely on simplifying assumptions. Our model consists of a shared encoder and two decoders for predicting the robot's centerline coordinates and its corresponding length from the base respectively. We benchmark the proposed approach against two baselines in a simulated dataset and show it outperforms both baselines by a large margin.

Index Terms—Continuum Robot, 3D Shape Sensing, Robot Vision

1 Introduction

Continuum Robot, as illustrated in Fig 1, refers to the subcategory of robotic manipulators that do not contain rigid links or identifiable joints. Due to their narrow curvilinear shape, structural compliance, and miniaturization capability, they have been researched for applications involving cluttered environments, such as minimally invasive surgery [1], non-destructive inspection [2], and space/sea exploration [3], [4].

Performing precise motion control of continuum robots requires real-time and accurate shape sensing. A direct way to calculate the shape of the continuum robot is to use a model-based method. However, the model-based method is sensitive to unknown external loads which leads to poor performance [5]. Another way to estimate the shape of the robot is to use additional sensors. Song and Wu et al. estimate the shape of the continuum robots using multiple electromagnetic sensors [6], [7]. The electromagnetic sensors are able to provide accurate position and direction information with respect to the global frame but the sensors take up valuable space in the robots and pose challenges to miniaturization.

Thus, we proposed a visual-based method for continuum robot shape sensing. Additionally, we approach visual shape estimation with potential application scenarios in mind – image from a single viewpoint at a time (monocular input) is almost always achievable (e.g., X-ray), but a stereo setup or depth camera may not be available. The purpose of the project is to investigate the feasibility of monocular visual shape estimation for a continuum robot in terms of accuracy and computation time. If successful, the method would efficiently close the control loop thus improving performance for many continuum robot applications.

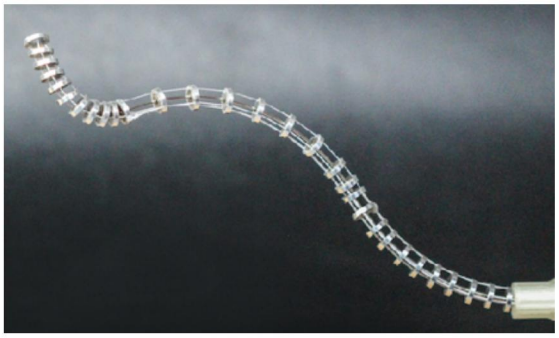


Fig. 1: Tendon-driven continuum robot prototype with three extensible sections at different lengths [8].

2 Related Work

2.1 Monocular 3D Object Reconstruction

Reconstructing 3D objects from a single viewpoint is known to be both challenging and ambiguous. To tackle this problem, early studies use an object shape prior to match the projected object silhouette with image cues [9]. These approaches are limited by the strong assumptions on lighting conditions and the use of a simplified model for surface reflectance [9].

In more recent attempts, learning-based approaches have become more popular as a result of the development of deep learning architectures. Fan et al. proposed a conditional generative model that can predict the 3D point cloud of an object from a single input image [10]. The model first conducts the encoding-decoding operations recurrently to learn surface details. Then the information gets passed into two parallel prediction branches - a fully-connected branch and a deconvolution branch. The fully-connected branch allows the model to describe intricate structures while the deconvolution branch is able to learn large smooth surfaces.

- *: *These authors contribute equally.*
- *T. Li is with the Department of Computer Science, University of Toronto, Canada.*
  E-mail: tli@cs.toronto.edu
- *J. Shentu is with the Department of Computer Science, University of Toronto, Canada.*
  E-mail: cshentu@cs.toronto.edu
- *V. Chen is with the Department of Mechanical Industrial Engineering, University of Toronto, Canada.*
  E-mail: chaojun.chen@mail.utoronto.ca

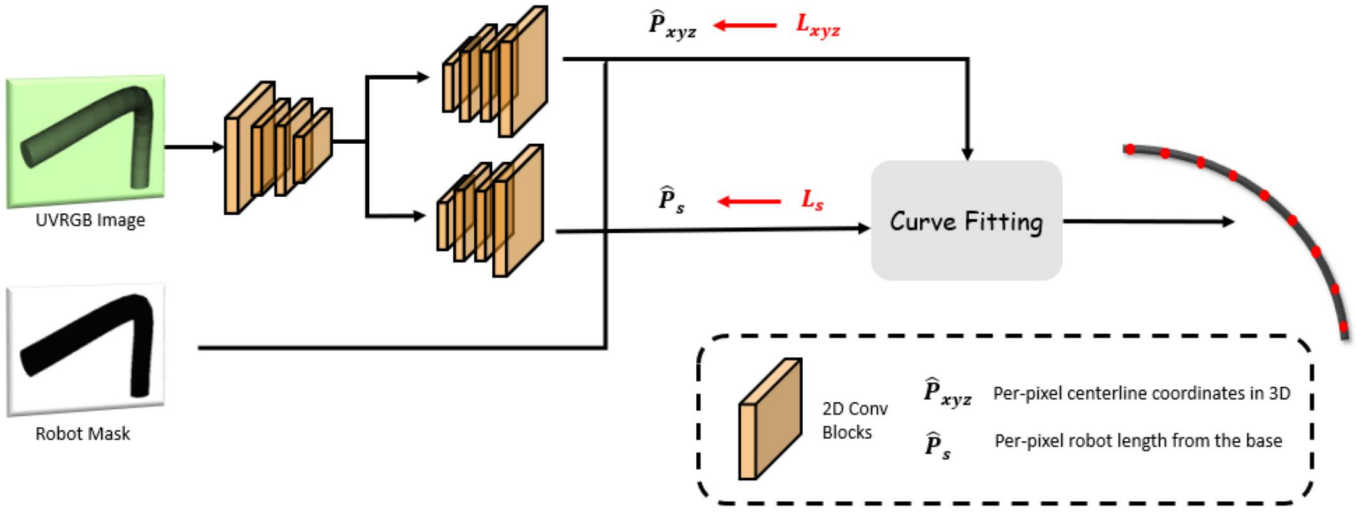


Fig. 2: Overview of the Framework

The outputs of the two branches are later merged together to form the final set of points [10].

In another research, Wang et al. designed an end-to-end deep learning architecture that produces a 3D shape in triangular mesh from a single color image [11]. The network, Pixel2Mesh, uses a graph-based convolutional neural network to represent 3D mesh and progressively deforms an ellipsoid to produce correct geometry. Compared to state-of-the-art approaches, Pixel2Mesh achieves higher 3D shape estimation accuracy and produces mesh models with better details [11].

2.2 3D Reconstruction of Continuum Robot

For 3D shape reconstruction of continuum robots, vision-based shape-sensing techniques have been validated to provide more direct and accurate measurement than solely using kinematic modeling [12].

One method is to first extract the robot/needle curve through segmentation and then estimate the 3D points for shape reconstruction using epipolar geometry analysis [13]. Burgner et al. achieved a mean error of 0.473 ± 0.353 mm on an anthropomorphic liver phantom with tumors and vessels [13]. Another method proposed by Dalvand et al. uses a stereo vision system and a 3D reconstruction algorithm based on the closed-form analytical solution for quadratic curve reconstruction in 3D space [14]. This method achieves real-time reconstruction of cardiac cathetersa with a maximum measurement error of 0.5 mm for the tip position and length and 0.5 degrees for the bending and orientation angles [14]. Croom et al. also uses a stereo vision based algorithm that employs self-organizing maps to triangulate 3D backbone curves from segmented 2D stereo images [15]. By avoiding the formation of 3D point cloud, the method is more efficient, and achieves a reconstruction accuracy of 1.53 mm at 4.0 Hz. [15].

While some of the results from previous research are promising in terms of accuracy, their suitability for real-life application is limited by slow speed, the requirement of multiple cameras or input images, and the dependency on tip- or body-mounted markers [14]. Additionally, there is a lack of common hardware or software benchmarks in the field, and some approaches are application specific or use simplifying assumptions to achieve good performance.

3 Method

We start by presenting the problem setup and then explain the proposed approach in detail. At a high level, we modify the UNet [16] architecture to have two decoders. Individually they predict 3D points on the centerline and its relative position on the robot. We apply the least-squares method to obtain a parametrized 3D curve describing the robot centerline.

3.1 Problem Formulation

Assume we are given an RGB image of the robot, $\mathbf{I}_{RGB} \in \mathbb{R}^{H\times W\times 3}$ and a binary occupancy mask of the robot, $O \in \mathbb{B}^{H\times W}$. The goal is to find the position of the robot in 3D, parameterized by the 3D coordinates of M evenly-spaced points on the centerline of the robot, denoted as $\mathbf{P}_r \in \mathbb{R}^{M\times 3}$.

The occupancy mask is assumed to come from an upstream segmentation module. The segmentation technique is highly dependent on the specific applications, so we will not include it in this work.

3.2 Network Architecture

We propose a network architecture that is based on UNet [16] but with two decoders. As depicted in Fig. 2, the network takes as input the UVRGB image of the robot constructed by appending the horizontal and vertical index (u, v) to the RGB value of each pixel. The encoder portion first applies repeated 3×3 convolution with batch normalization and a rectified linear unit (ReLU) for an initial feature map with 64 channels. Afterward, there are 4 downsampling steps using 2×2 max pooling operation with stride 2, followed by repeated convolutions described above to double the number of feature channels at each step. This brings the input image with dimension $512 \times 512 \times 5$ to a feature map of size $64 \times 64 \times 1024$. The two decoders

have the reverse structure and use bilinear interpolation to up-sample the feature maps. The encoder feature map before every down-sampling step is also concatenated to the feature map after every up-sampling step. In the last step, we use 1×1 convolution to output the final features for each pixel.

We use the binary occupancy mask to filter the outputs to obtain $\hat{\mathbf{P}}_{xyz} \in \mathbb{R}^{N\times 3}$ and $\hat{\mathbf{P}}_{\mathbf{s}} \in \mathbb{R}^{N}$ from the two decoders respectively, where N is the number of pixels belong to the robot. $\hat{\mathbf{P}}_{xyz}$ is a point cloud of the predicted robot's centerline. Intuitively, the first decoder is learning some depth information for each pixel, as well as the camera model to project that depth information into the robot's frame of reference. $\hat{\mathbf{P}}_{\mathbf{s}}$ is a parameterization variable ranging from 0 to 1 for each pixel. The variable represents each pixel's relative location with respect to the robot — 0 means it's at the robot's base and 1 means it's at the robot's tip.

Since the two decoders output a 3D point and a predicted parametrization for each pixel, we can conveniently apply the least-squares method to fit a 3D polynomial curve with arbitrary degrees.

3.3 Supervision

Using the depth image $\mathbf{I}_{depth} \in \mathbb{R}^{H\times W}$ and robot centerline points $\mathbf{P}_r \in \mathbb{R}^{M\times 3}$, we construct the ground truth point cloud $\mathbf{P}_{xyz} \in \mathbb{R}^{N\times 3}$ and $\mathbf{P}_s \in \mathbb{R}^{N}$ as follows. Points in $\mathbf{P}_r$ are ordered and equally spaced along the robot, so every point corresponds to a ground truth parametrization:

$$\mathbf{P}_{sr,j} = \frac{j-1}{M}, \quad j = 1, ..., M$$

For every point in the binary occupancy mask, we use the invertible camera projection matrix $\mathbf{C} \in \mathbb{R}^{4\times 4}$ and depth image to obtain the 3D point cloud corresponding to each pixel.

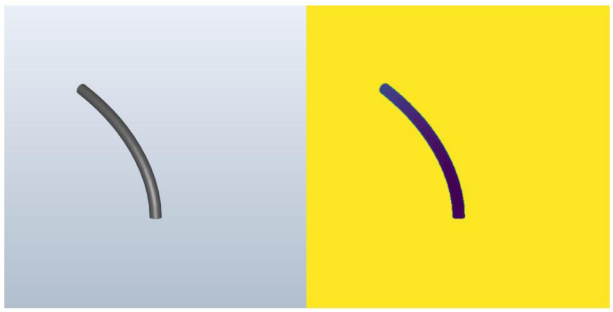
$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{C}^{-1} \begin{bmatrix} u \\ v \\ 1 \\ 1/d \end{bmatrix}$$

where (u, v) is the index of the pixel on the image and d is its corresponding depth value from $\mathbf{I}_{depth}$. This gives us a ground truth point cloud on the robot surface captured by the camera, denoted $\mathbf{P}_{surface} \in \mathbb{R}^{N\times 3}$. For each point on the surface, we find the closest centerline point coordinates and the parameterization values, $\mathbf{P}_{xyz} \in \mathbb{R}^{N\times 3}$ and $\mathbf{P}_s \in \mathbb{R}^{N}$, respectively. We supervise the network using the loss function depicted as follows.

$$L = \beta_1 \|\hat{\mathbf{P}}_{xyz} - \mathbf{P}_{xyz}\|_2 + \beta_2 \|\hat{\mathbf{P}}_x - \mathbf{P}_x\|_2$$

4 Experiment

We train and evaluate the proposed model on a custom dataset collected from simulation. Common metrics used in continuum robot research are adopted to measure the accuracy of shape sensing and tip tracking. We then present the baselines adopted as well as the experimental settings.



(a) RGB image (b) depth image

Fig. 3: Sample Image Data from Custom Dataset

4.1 Dataset

We collected a custom dataset using an existing simulator. The simulated tendon-driven continuum robot is 280 mm in length and 10 mm in radius with a protective sleeve. Randomly sampled robot configurations are rendered with the Visualization Toolkit (VTK), where we save 512×512 RGB and depth images (Fig. 3) along with camera configuration and ground truth robot shape. The dataset contains 50,000 robot configurations. Texture was added to make the dataset more realistic. 80% of the dataset are for training and validation, and the remaining 20% are reserved for testing.

4.2 Metrics

Shape sensing for continuum robots has typically been evaluated in terms of mean error of robot shape (**MERS**) and mean error of tip tracking (**METE**) [17]. Although they have been calculated differently across literature, we define **MERS** to be the average Euclidean distance between the predicted set of evenly-spaced points, $\hat{\mathbf{P}}_r \in \mathbb{R}^{M\times 3}$, and corresponding ground truth points, $\mathbf{P}_r \in \mathbb{R}^{M\times 3}$, across different configurations in the robot's workspace.

$$\mathbf{MERS} = \frac{1}{M}\sum_{j=1}^{M} \|\hat{\mathbf{P}}_{r,j} - \mathbf{P}_{r,j}\|_2$$

We also constraint $M \geq 10$ so the points are representative of the robot's overall shape. **METE** is calculated in the same way but only accounting for the tip position.

$$\mathbf{METE} = \|\hat{\mathbf{P}}_{r,M} - \mathbf{P}_{r,M}\|_2$$

We evaluate our method against these two metrics with and without external loading to better reflect application scenarios.

4.3 Baselines

We use a combination of UNet [16] and PointNet [18] as the first baseline where a captured image of the robot is processed by the UNet to obtain per-pixel 3D projection. The PointNet then takes the 3D point cloud of the robot and outputs the coordinates of the points on the robot centerline.

Further, we combine UNet with a 6-degree polynomial fitting algorithm for the second baseline. After decoding per-pixel centerline coordinates from the UNet, the curve fitting algorithm fits three 2-dimensional curves on x, y, and z axis with respect to the curve length. The curve length is approximated using the distance from the robot base.

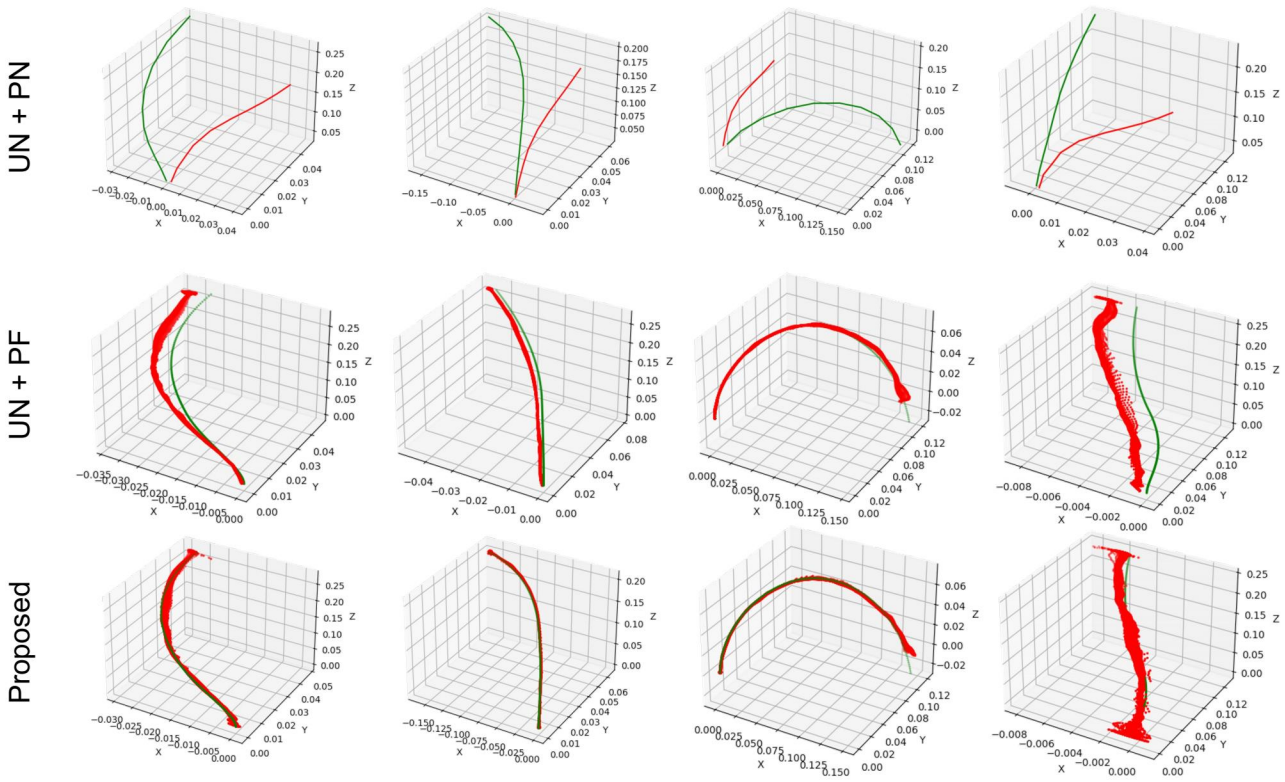


Fig. 4: Qualitative comparison of the proposed approach against the two baselines. **UN**: Unet. **PN**: PointNet. **PF**: polynomial fitting. red curve: network output. green curve: ground truth robot centerline.

4.4 Experimental Setup

We trained the network end-to-end for 50 epochs using AdamW optimizer with learning rate of 0.003. A weight decay of 10^{-4} was used. The model was trained on a single NVIDIA RTX2080Ti with a batch size of 4 per GPU. β_1 and β_2 were both set to 1.

5 Results

In this section, we present the quantitative and qualitative results of the proposed approach on the simulated dataset and provide an ablation analysis on various components of the model.

5.1 Quantitative Results

We summarize the quantitative results of the model on the test split of the simulated dataset in TABLE 1. Compare to the two baselines, our approach obtains significantly lower errors on both **MERS** and **METE** while is still able to run in real-time.

Model	MERS	METE	FPS
UNet [16] + PointNet [18]	16.22	34.47	26.43
UNet [16] + Polynomial Fitting	15.74	15.94	26.81
Ours	**1.74**	**3.09**	**20.34**

TABLE 1: Test set result on the simulated dataset. Metrics are in terms of mm. FPS are in terms of Hz.

5.2 Qualitative Results

We present sample model outputs visually in Fig. 4. The first baseline (Row 1: UN + PN) regresses M evenly-spaced points on the robot centerline directly, yet its predictions are prone to errors. Continuum robots do not have clear joints or key points visually. Thus, it is quite challenging to regress coordinates of desired locations on the robot directly. On the other hand, using the combination of UNet and polynomial fitting (Row 2), the model is able to predict reasonably well for the easier poses. However, it could not handle cases of complex robot configurations (Col 4). Clearly, the proposed approach (Row 3) yields the best results.

5.3 Ablation Analysis

We conduct ablation studies to analyze how each of the components designed contributes to the final results. Specifically, we study how multiple decoders could help the network in learning. M1 and M2 in TABLE 2 correspond to the two baselines presented earlier for easier referencing. Moving from M2 to M3, we experiment separating the centerline coordinates learning by first decoding the coordinates on the robot surface and then regressing the offset towards the center. However, we see the error increases when separating this task into two sub-tasks thus decide to keep the network compact. Additionally, comparing M2 and M7 (likewise for comparing M3 and M6), we see a drastic decrease in both **MERS** and **METE** when adding an extra decoder to learn the per-pixel length of the point from the robot base. This value is then used for curve fitting to replace the Euclidean distance from the base used by M2.

Note that results of M7 slightly differs from TABLE 1 due to different polynomial degrees in curve fitting. We use 4-th order polynomial fitting to compile the best results in TABLE 1 but keep the setting the same as baseline (6-th order) in TABLE 2 for fair comparison. We present in depth how the polynomial degree affects the results in TABLE 3.

Architecture		Backbone			Decoders				Fitting		Metrics		
		UN	RN	SN	D1	D2	D3	D4	PN	PF	MERS	METE	FPS
Baseline	M1	✓			✓				✓		16.22	34.47	26.43
	M2	✓				✓				✓	15.74	15.94	26.81
Ablations	M3	✓			✓		✓			✓	15.93	17.45	20.12
	M4		✓		✓		✓			✓	15.40	16.59	11.01
	M5			✓	✓		✓			✓	15.73	17.54	15.94
	M6	✓			✓		✓	✓		✓	1.79	3.48	15.87
Proposed	M7	✓				✓		✓		✓	**1.76**	**3.29**	**20.34**

TABLE 2: Ablation study of the proposed components vs baseline. **UN**: UNet [16], **RN**: ResNet [19], **SN**: SalsaNext [20], **D1**: Decoding per-pixel xyz coordinates on surface, **D2**: Decoding per-pixel xyz coordinates on centerline, **D3**: Decoding per-pixel Δxyz offset from surface to centerline, **D4**: Decoding per-pixel length from robot base, **PN**: PointNet, **PF**: Polynomial fitting with degree of 6. **FPS**: Frames per second measured using Intel(R) Xeon(R) CPU E5-2687W v4 and NVIDIA RTX 2080Ti. Metrics are presented in mm.

Further, we show how changing the backbone of the network affects the model performance. We conduct experiments on changing the backbone from the Vanilla UNet (M3) [16] to ResNet (M4) [19] and SalsaNext (M5) [20]. The results are summarized in Rows 3-5 in TABLE 2. Overall, the model is quite robust against the change of backbone. ResNet obtains slightly better results while trading off quite a lot of computation. We argue that doubling the runtime here is not worth to gain about 0.5 mm in **MERS**. However, in cases where runtime is not a concern, one is encouraged to explore more with ResNet backbone.

Lastly, we investigate how the degree number in polynomial fitting affects the results. We conduct experiments by fitting polynomials of various degrees on the network output (per-pixel robot centerline projected in 3D) and present the summary of results in TABLE 3. Using polynomial of degree 4 yields the best results, and we start to see overfitting when it's degree of 5 or higher.

Polynomial Degree	MERS	METE
2	2.95	5.12
3	1.77	3.16
4	**1.74**	**3.09**
5	1.75	3.18
6	1.76	3.29

TABLE 3: Ablation analysis on polynomial degree

6 Conclusion

We presented a monocular approach to provide shape sensing for the *Continuum robots*. The model takes as input the RGB image and the binary segmentation mask of the robot and predicts a list of 3D coordinates describing its 3D location in the scene. We evaluated the proposed method in a simulated dataset and presented quantitative and qualitative results showing that it outperforms the baselines by a large margin. Future work will include the extension of our method to the real-world dataset and investigate the sim-to-real transfer learning.

References

[1] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015. 1

[2] X. Dong, D. Axinte, D. Palmer, S. Cobos, M. Raffles, A. Rabani, and J. Kell, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 218–229, 2017. 1

[3] I. D. Walker, "Continuum robot appendages for traversal of uneven terrain in in situ exploration," in *Aerospace Conference*, 2011, pp. 1–8. 1

[4] M. Sfakiotakis, A. Kazakidi, N. Pateromichelakis, J. A. Ekaterinaris, and D. P. Tsakiris, "Robotic underwater propulsion inspired by the octopus multi-arm swimming," in *IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3833–3839. 1

[5] J. Li, F. Zhang, Z. Yang, Z. Jiang, Z. Wang, and H. Liu, "Shape sensing for continuum robots by capturing passive tendon displacements with image sensors," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, p. 3130–3137, 2022. 1

[6] S. Song, Z. Li, M. Q.-H. Meng, H. Yu, and H. Ren, "Real-time shape estimation for wire-driven flexible robots with multiple bending sections based on quadratic bézier curves," *IEEE Sensors Journal*, vol. 15, no. 11, p. 6326–6334, 2015. 1

[7] L. Wu, S. Song, K. Wu, C. M. Lim, and H. Ren, "Development of a compact continuum tubular robotic system for nasopharyngeal biopsy," *Medical amp; Biological Engineering amp; Computing*, vol. 55, no. 3, p. 403–417, 2016. 1

[8] M. Neumann and J. Burgner-Kahrs, "Considerations for follow-the-leader motion of extensible tendon-driven continuum robots," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016. 1

[9] P. Guan, A. Weiss, A. O. Balan, and M. J. Black, "Estimating human shape and pose from a single image," *2009 IEEE 12th International Conference on Computer Vision*, 2009. 1

[10] H. Fan, H. Su, and L. Guibas, "A point set generation network for 3d object reconstruction from a single image," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[11] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2mesh: Generating 3d mesh models from single rgb images," *Computer Vision – ECCV 2018*, p. 55–71, 2018. 2

[12] C. Shi, X. Luo, P. Qi, T. Li, S. Song, Z. Najdovski, T. Fukuda, and H. Ren, "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, p. 1665–1678, 2017. 2

[13] J. Burgner, S. D. Herrell, and R. J. Webster, "Toward fluoroscopic shape reconstruction for control of steerable medical devices," *ASME 2011 Dynamic Systems and Control Conference and Bath/ASME Symposium on Fluid Power and Motion Control, Volume 2*, 2011. 2

[14] M. M. Dalvand, S. Nahavandi, and R. D. Howe, "High speed vision-based 3d reconstruction of continuum robots," *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016. 2

[15] J. M. Croom, D. C. Rucker, J. M. Romano, and R. J. Webster, "Visual sensing of continuum robot shape using self-organizing maps," *2010 IEEE International Conference on Robotics and Automation*, 2010. 2

[16] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, ser. LNCS, vol. 9351. Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). [Online]. Available: http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a 2, 3, 4, 5

[17] C. Shi, X. Luo, P. Qi, T. Li, S. Song, Z. Najdovski, T. Fukuda, and H. Ren, "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," vol. 64, no. 8, pp. 1665–1678, conference Name: IEEE Transactions on Biomedical Engineering. 3

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660. 3, 4

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. 5

[20] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast semantic segmentation of lidar point clouds for autonomous driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 655–661. 5