MoSS: Monocular Shape Sensing for Continuum Robots

Chengnan Shentu, *Graduate Student Member, IEEE*, Enxu Li, *Graduate Student Member, IEEE*, Chaojun Chen, Puspita T. Dewi, David B. Lindell, *Member, IEEE*, and Jessica Burgner-Kahrs, *Senior Member, IEEE*

***Abstract*—Continuum robots are promising candidates for interactive tasks in medical and industrial applications due to their unique shape, compliance, and miniaturization capability. Accurate and real-time shape sensing is essential for such tasks yet remains a challenge. Embedded shape sensing has high hardware complexity and cost, while vision-based methods require stereo setup and struggle to achieve real-time performance. This letter proposes a novel eye-to-hand monocular approach to continuum robot shape sensing. Utilizing a deep encoder-decoder network, our method, MoSSNet, eliminates the computation cost of stereo matching and reduces requirements on sensing hardware. In particular, MoSSNet comprises an encoder and three parallel decoders to uncover spatial, length, and contour information from a single RGB image, and then obtains the 3D shape through curve fitting. A two-segment tendon-driven continuum robot is used for data collection and testing, demonstrating accurate (mean shape error of 0.91 mm, or 0.36% of robot length) and real-time (70 fps) shape sensing on real-world data. Additionally, the method is optimized end-to-end and does not require fiducial markers, manual segmentation, or camera calibration.**

***Index Terms*—Modeling, control, and learning for soft robots, computer vision for medical robotics, data sets for robot learning.**

I. Introduction

CONTINUUM robots are robotic manipulators that do not contain rigid links or identifiable joints. They have been studied for interactive applications such as minimally invasive surgery [1] and non-destructive inspection [2]. To enable continuum robots to perform these tasks in a flexible and adaptable manner, accurate and real-time 3D shape sensing is crucial—by tracking the robot's shape in the environment, controllers can be applied to reach a desired position, avoid collisions, or follow a certain path. Additionally, shape sensing allows the monitoring of the robot's condition and performance.

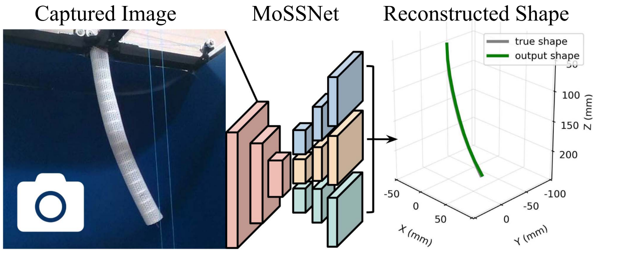


Fig. 1. Our method, MoSSNet, takes a single camera image as input and outputs an accurate parametric representation of the robot centerline in real-time, without requiring fiducial markers, manual segmentation or camera calibration.

Model-based shape reconstruction is proposed to estimate the 3D shape of continuum robots from internal sensors in the drive-system and actuators, but they have a trade-off between accuracy and computation cost [3]. Additionally, they are sensitive to uncertainty in the modeling parameters, unmodelled effects, and unknown external loads. To address these limitations, sensing-based approaches have been proposed, which can be broadly categorized as embedded or vision-based methods.

Embedded sensors, such as fiber-optic sensors, electromagnetic (EM) sensors, and force/torque sensors, provide indirect measurements for shape reconstruction, but they require customized integration efforts and can be sensitive to external interference [4]. Vision-based methods, on the other hand, offer high accuracy at low cost and can be easily adapted to different robots. However, existing methods require calibrated, multi-view camera systems, and most are not capable of real-time applications, limiting their application outside a lab environment [5].

This tradeoff between hardware complexity and performance motivates a data-driven monocular approach that offers accurate sensing at low cost while mitigating the performance overhead of markerless stereo matching. Our method, MoSSNet, achieves this objective utilizing a deep encoder-decoder network as illustrated in Fig. 1.

II. Related Work

In this section, we discuss the diverse approaches taken for 3D shape sensing of continuum robots. Based on the sensing

Manuscript received 27 June 2023; accepted 8 December 2023. Date of publication 22 December 2023; date of current version 5 January 2024. This letter was recommended for publication by Associate Editor P. Maiolino and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grant RGPIN-2019-04846. *(Chengnan Shentu and Enxu Li are co-first authors.) (Corresponding author: Chengnan Shentu.)*

Chengnan Shentu, Puspita T. Dewi, and Jessica Burgner-Kahrs are with the Continuum Robotics Laboratory, Department of Mathematical and Computational Sciences, University of Toronto, Mississauga, ON L5L1C6, Canada (e-mail: cshentu@cs.toronto.edu; puspita.dewi@utoronto.ca; jessica.burgner-kahrs@utoronto.ca).

Enxu Li is with the Department of Computer Science, University of Toronto, Toronto, ON M5S1A1, Canada, and also with Waabi Innovation Inc., Toronto, ON M5T2S6, Canada (e-mail: tli@cs.toronto.edu).

Chaojun Chen is with the Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S1A1, Canada (e-mail: chaojun.chen@mail.utoronto.ca).

David B. Lindell is with the Department of Computer Science, University of Toronto, Toronto, ON M5S1A1, Canada (e-mail: lindell@cs.toronto.edu).

Code and datasets are available at https://github.com/ContinuumRoboticsLab/MoSSNet.

Digital Object Identifier 10.1109/LRA.2023.3346271

2377-3766 © 2023 Crown Copyright

TABLE I
OVERVIEW OF METHODS FOR CONTINUUM ROBOT SHAPE SENSING

Sensing method	Error (mm)	Update rate (fps)	Hardware complexity	Line-of-sight requirement	References
Electromagnetic tracking	low (0.5-1.5 or 0.5-1.5%)	medium to high (10-120)	high	none	[6], [7]
Optical strain sensor	very low ($\leq$ 0.5 or 0.5%)	high (60-120)	high	none	[8], [9]
Force/torque sensor	high ($\geq$ 3 or 3%)	very high ($\geq$120)	medium	none	[10]
Stereo vision	low	low ($\leq$10)	low	two	[11], [12], [13]
Monocular vision (Ours)	low	high	low	one	

Sensors without line of sight requirements have high hardware complexity/cost, while stereo methods tend to be slow and require line of sight to multiple cameras. Our method is low-cost, accurate, efficient, and requires line of sight to only a single camera.

principle, we categorize them into embedded and vision-based methods and discuss them separately. A brief overview of these approaches is presented in Table I.

A. Embedded Shape Sensing

Magnetic sensors, typically used in electromagnetic (EM) tracking systems, can be attached to the robots and localized by measuring the small current induced by a magnetic field generated in the workspace. Thus, such methods are sensitive to EM interference and have limited workspace, which poses constraints on the robot material and application environments [4]. Because each sensor only provides pose information at a single point, determining the number of sensors to use is crucial. A higher number of sensors can lead to increased accuracy, but rigid sensors can interfere with the continuum robot's characteristics. Conversely, a lower number of sensors are more efficient, but rely more heavily on kinematic models [7] or shape representations [6], which can make the process more computationally expensive and vulnerable to model uncertainties.

Strain sensors, such as optical fibers with fiber Bragg gratings (FBGs), measure shape by integrating local curvature/strain. Their small size and bio-compatibility allow embedded shape sensing in varied environments with real-time capability and has motivated research and commercialization efforts [8], [9]. However, they require customized sensor setups for different robots, which leads to high costs [14].

Lower-cost alternatives such as passive strings [15] have been proposed at the expense of lower accuracy and update rate. Orekhov et al. [16] formally analyzed the problem of string routing optimization through sensitivity analysis, and achieved tip error of 5.9 mm(1.97%) at 4 fps. Moreover, they share the same drawback of being highly sensitive to the placement and calibration of the sensors.

In addition to electromagnetic tracking and strain-based shape sensing, force/torque sensors have been employed to estimate the shape of elastic rods using the Kirchhoff rod model [17]. This method is cost-effective and computationally efficient. While attempts have been made to extend it to multi-segment continuum robots, position errors increase with larger deformation due to unmodeled effects [10].

Common to all embedded shape sensing approaches is they do not require line-of-sight, which is valuable for applications in confined spaces. However, they have a higher level of hardware complexity because they require customized sensor integration and calibration efforts. Their accuracy and update rate scale with the performance of sensing hardware, which also leads to a high cost of deployment (see Table I).

B. Vision-Based Shape Sensing

Vision-based eye-to-hand shape sensing has gained attention since they are cost effective and have low hardware complexity. One of the most straightforward approaches is detecting point correspondence in stereo setups. Delmas et al. achieved mean error of 0.46 mm (2.30%) at 6.25 fps in simulated fluoroscopic image pairs [12]. Such methods can be slow for continuum robots, because they lack identifiable joints and links as feature points.

Fiducial markers can be added to speed up point correspondence [18], but integrating markers that meet size, shape, and visibility constraints is difficult in applications. Another approach to speed up stereo shape sensing is to apply simplifying assumptions, such as limiting the shapes to quadratic forms, as done by Dalvand et al. [19]. However, this approach does not generalize to robots capable of more complex shapes. Camarillo et al. [20] proposed 3D shape reconstruction using the shape-from-silhouette technique with three cameras. Self-organizing maps algorithm is also investigated by Croom et al. [13] to fit a 3D curve from point cloud. Although these methods are accurate, and achieve mean error of 1.14 mm(0.72%) and 1.53 mm(0.64%) respectively, they still require point correspondence and are computationally expensive ($\leq$ 4 fps).

C. Monocular Computer Vision

At its core, vision-based shape sensing requires some depth information on the robot to reconstruct its 3D shape, which is traditionally achieved through epipolar geometry with two or more cameras. More recently, learning-based monocular methods has gained success in the domain of depth estimation [21] and 3D object detection [22] for self-driving, as well as 3D human pose estimation [23]. Most existing methods consist of encoder-decoder-based or transformer-based architectures, which emphasize on local and global features respectively. Since continuum robots are unique in terms of their jointless body, we focus on encoder-decoder architecture to capture local curvature information.

Our approach, MoSSNet, achieves accurate (mean shape error of 0.91 mm(0.36%)) and real-time (70 fps) results on a real-world dataset. The method takes a single RGB image as input and outputs a parametric representation of the robot centerline,

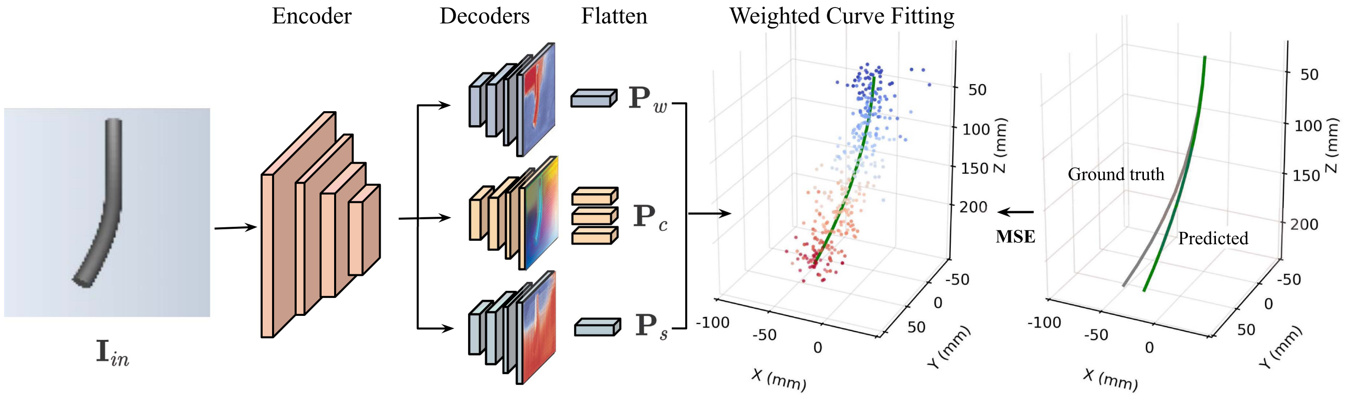


Fig. 2. Overview of our approach, MoSSNet. The network takes as input the captured image of the robot and generates importance for reconstruction, centerline coordinates, and relative arclength. These flattened outputs are then processed by the weighted curve fitting algorithm to generate a curve that parameterizes the robot's centerline. To train the network, we supervise its learning process by penalizing the mean squared error between the predicted and ground truth curves.

without requiring fiducial markers, manual segmentation, or camera calibration.

III. MoSSNet

In this section, we introduce Monocular Shape Sensing Network (MoSSNet), an efficient and effective approach for continuum robot shape estimation. The problem formulation, network architecture, and training methodology are introduced in the following.

A. Problem Formulation

We suppose there is a camera that remains in a fixed pose relative to the robot's base. The camera captures an image of the robot, which we refer to as $\mathbf{I}_{RGB} \in \mathbb{R}^{H\times W\times 3}$ where H and W are the height and width of the captured image. The objective is to determine the 3D centerline of the robot, which is the widely adopted representation of continuum robots' shape [5]. Specifically, we identify the coordinates of M equally spaced points along its centerline, represented as $\mathbf{P}_r \in \mathbb{R}^{M\times 3}$.

In this work, we limit the scope to monocular RGB images with uniform lighting conditions and no occlusions except the robot's self-occlusions. We also assume the robot's geometry remains constant and its motion generates minimal to no motion blur on the images captured.

B. Network Architecture

In order to reconstruct the robot, our model uses an image of the robot to predict the 3D coordinates along its centerline. Subsequently, a weighted linear least squares algorithm is employed to derive a 3D curve that parametrizes the center of the robot. As shown in Fig. 2, we design a network with a shared encoder and three decoders that are composed of four stages each. These stages consist of a residual block [24] with two convolutional layers that are connected through BatchNorm [25] and Leaky ReLU activations [26]. The encoder block uses a maxpooling layer between every two stages to decrease the feature map's size by a factor of two. In contrast, the decoder block incorporates a pixel shuffle layer [27] between every two stages to increase the feature map's size by a factor of two. Next, we explain each component in more detail.

a) Encoder: To incorporate location information, we add the 2D image indices to $\mathbf{I}_{RGB}$. This results in a 5-channel image, represented as $\mathbf{I}_{in} \in \mathbb{R}^{H\times W\times 5}$. The encoder then extracts multi-scale features from the input image, which are subsequently passed through three decoders for further processing.

b) Decoders: Given that the image includes background, not every pixel is relevant in determining the robot's shape. To address this, the importance decoder learns the significance of each pixel in shape reconstruction. The output of the importance decoder is a per-pixel importance score denoted as $\mathbf{P}_w \in \mathbb{R}^{HW}$, which is normalized between 0 and 1 using a Sigmoid function. A higher value of this score indicates that the pixel is more relevant for shape sensing, whereas a lower value suggests that it belongs to the background. These scores will be used to perform weighted curve fitting in later stages.

Subsequently, we use another decoder to generate an estimate of the robot's shape by predicting its xyz coordinates along its centerline, denoted as $\mathbf{P}_c \in \mathbb{R}^{HW\times 3}$. It is worth noting that only regions that cover the robot's shape, i.e., regions with high importance scores, will have useful values. Further, the last decoder learns a per-pixel relative arclength that ranges from 0 to 1, where 0 represents the robot's base and 1 represents its tip, denoted by $\mathbf{P}_s \in \mathbb{R}^{HW}$.

c) Weighted Curve Fitting: We model the robot centerline as three independent n-th order polynomials on x, y, and z axes with curve parameters being $\mathbf{w} = [\mathbf{w}_x \ \mathbf{w}_y \ \mathbf{w}_z] \in \mathbb{R}^{(n+1)\times 3}$. Formally, we have

$$\underbrace{\begin{bmatrix} | & | & & | \\ (\mathbf{P}_s)^0 & (\mathbf{P}_s)^1 & \dots & (\mathbf{P}_s)^n \\ | & | & & | \end{bmatrix}}_{\mathbf{A}\in\mathbb{R}^{HW\times(n+1)}} \underbrace{\begin{bmatrix} | & | & | \\ \mathbf{w}_x & \mathbf{w}_y & \mathbf{w}_z \\ | & | & | \end{bmatrix}}_{\mathbf{w}\in\mathbb{R}^{(n+1)\times 3}}$$

$$= \underbrace{\mathbf{P}_c}_{\mathbf{B}\in\mathbb{R}^{HW\times 3}} . \quad (1)$$

To obtain a curve of best fit, we compute the weighted least squares solution of the curve parameter $\mathbf{w}$ as follows,

$$\mathbf{w} = \left(\mathbf{A}^T\mathbf{\Sigma}\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{\Sigma}\mathbf{B}, \tag{2}$$

where $\mathbf{\Sigma} = \text{diag}(\mathbf{P}_w) \in \mathbb{R}^{HW\times HW}$ is the learned per-pixel weighting for the curve fitting. Finally, we obtain the coordinates of the M evenly-spaced points along the robot centerline by querying M relative locations.

$$\underbrace{\begin{bmatrix} (1/M)^0 & (1/M)^1 & \dots & (1/M)^n \\ (2/M)^0 & (2/M)^1 & \dots & (2/M)^n \\ \vdots & & & \\ (M/M)^0 & (M/M)^1 & \dots & (M/M)^n \end{bmatrix}}_{\mathbf{P}_q\in\mathbb{R}^{M\times(n+1)}} \mathbf{w} = \underbrace{\begin{bmatrix} | & | & | \\ \hat{\mathbf{p}}_x & \hat{\mathbf{p}}_y & \hat{\mathbf{p}}_z \\ | & | & | \end{bmatrix}}_{\hat{\mathbf{P}}_r\in\mathbb{R}^{M\times 3}}. \tag{3}$$

C. Supervision

We use the weighted mean squared error of the M predicted coordinates and the ground truth as the loss function to supervise the network. The loss function is depicted as follows,

$$L = \frac{1}{M}\sum_{j=1}^{M}\beta_j\|\hat{\mathbf{P}}_{r,j} - \mathbf{P}_{r,j}\|_2^2, \tag{4}$$

where β_j is the weight applied on each of the M points. Given the inherent difficulty in accurately reconstructing the tip location of the robot, we assign a larger weight on β_M to penalize the tip error.

IV. Data Collection and Benchmark

The monocular shape sensing method proposed is evaluated both in simulation and on a robot prototype. While the method can be applied to all types of continuum robots, we focus on tendon-driven continuum robots (TDCRs), which represent one of the most extensively utilized and studied varieties of continuum robots.

First, we describe the hardware and simulation setup for data collection. Afterward, we provide an overview of our dataset and define the metrics used for benchmarking.

A. Hardware Setup

For this work we use a TDCR that is 250 mm in length and 20 mm in diameter. It has two identical bending segments with 20 equally distributed spacer disks for tendon routing along a super-elastic Nitinol backbone. The robot is covered in a flexible polyethylene sleeve. The TDCR is mounted on a mechanical frame and tendons are manually operated.

We collect RGB and depth images using an RGB-D camera (RealSense D415, Intel, USA) at 1280×720 resolution. The ground truth shape of the TDCR is measured by an FBG shape sensing system (custom multicore fiber, fan-out box, and FBG-Scan 908 interrogator, FBGS International NV, Belgium) placed inside the backbone. The multicore fiber sensor has sensing length of 250 mm, which contains 26 evenly spaced gratings, and outputs a shape measurement of 251 points (with RMS error of 1.27 mm or 0.5%). During data collection, the camera allows simultaneous RGB and depth image capture at 6 Hz, while the FBG system provides updates at 100 Hz. We collect the shape measurement with the closest timestamp for each camera frame, resulting in a worst-case time difference of 5 ms.

The FBG sensor is positioned such that its coordinate frame aligns with that of the robot. During training, the RGB images are inputs to the network. Points along the robot centerline are given directly by the FBG system and are used as ground truth. Depth images are not used in our method, but are included in the dataset to facilitate further research.

B. Simulation Setup

A TDCR with the same parameters is simulated using the Cosserat rod-based static model [28], specifically a C++ implementation published in [29]. The simulated robot is then rendered using the Visualization Toolkit (VTK) with simple texture and lighting [30]. We apply calibrated camera intrinsic and extrinsic from the hardware setup to obtain the same view, as shown by the sample images in Fig. 3. By sampling random joint space configurations, we simulate realistic robot shapes and obtain RGB and depth images from VTK. The model implementation also allows us to sample robot shapes under external load, so we also simulate robot shapes with a random external force and moment at the tip.

For training, the RGB images are inputs to the network. Points along the robot's centerline, provided by the simulator, are used as the ground truth.

C. MoSS-Real and MoSS-Sim Datasets

We present our dataset for monocular shape sensing of a two-segment TDCR. Using the hardware and simulation setup outlined above, we collect 12 000 configurations on the robot prototype and 50 000 configurations rendered in the simulator, including 25 000 in free space and 25 000 with external load at the tip with 3 force components and 3 moment components. Each force component is randomly sampled between $[-0.1\,\text{N}, 0.1\,\text{N}]$, and each moment component is randomly sampled between $[-0.01\,\text{Nm}, 0.01\,\text{Nm}]$. We will refer to the two datasets as the MoSS-Real dataset and the MoSS-Sim dataset in later sections of this letter. We split both datasets into training, validation, and test sets randomly with ratios of 60%, 15%, and 25% respectively. Depth images and camera parameters are not used in our proposed method but are included to support the development of alternative methods.

D. Benchmarking Metrics

Shape sensing for continuum robots is typically evaluated in terms of mean error of robot shape (MERS) and mean

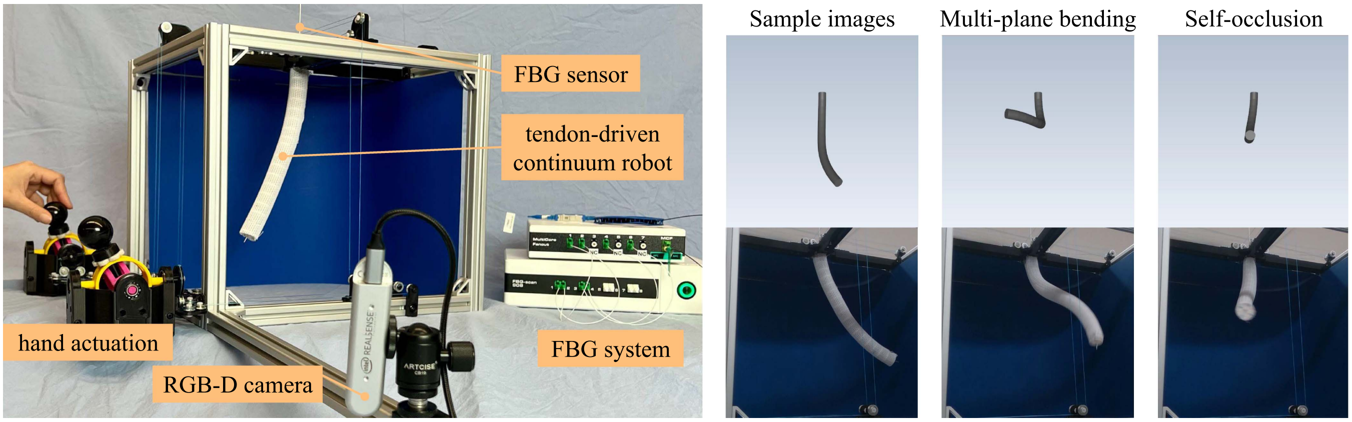


Fig. 3. We collect a monocular shape sensing dataset with a two-segment tendon-driven continuum robot on hardware and in simulation. An overview of hardware setup is shown on the left, and sample images are shown on the right.

error of robot tip (MERT) [5]. 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.

$$\text{MERS} = \frac{1}{M}\sum_{j=1}^{M} \|\hat{\mathbf{P}}_{r,j} - \mathbf{P}_{r,j}\|_2. \tag{5}$$

Shape sensing output should provide information dense enough to reconstruct the complete robot shape for applications like collision checking. Thus, the minimum number of output points depend on the complexity of robot shape representation (i.e., degree-of-freedom in configuration space), such that there are enough data points for model fitting. We further constraint M to be at least double the minimum number of points to take account for errors from robot shape representation and avoid aliasing. In our case, 4th order polynomial is used and requires 5 points for curve fitting, so we constraint $M \geq 10$. MERT is calculated in the same way but only accounting for the tip position.

$$\text{MERT} = \|\hat{\mathbf{P}}_{r,M} - \mathbf{P}_{r,M}\|_2. \tag{6}$$

To facilitate comparison of results between different robots, we also report MERS and MERT with respect to the sensed length of the robot as a percentage. In our case the entire TDCR is being sensed, so we divide the error by 250 mm. We also evaluate our method's real-time capability by reporting its update rate in frames per second (fps). We evaluate our method against these three metrics on a large number of different shapes to ensure its robustness.

V. Evaluation

In this section, we first present the implementation details and then show the quantitative and qualitative results of the proposed approach on MoSS-Sim and MoSS-Real datasets. Further, we explore the approach's sim-to-real transfer capability and provide an ablation analysis on various components of the network. Finally, we discuss the robustness of our approach when the camera configuration changes.

A. Implementation Details

To process the captured image, we first crop it to 512 × 512 using a manually defined region of interest. We then use nearest downsampling to scale the image to 128 × 128 before feeding it into our network. We use a fourth order polynomial to model the centerline of the robot, and take $M = 10$. During training, we use a batch size of 4 and run our experiments on an NVIDIA T4 GPU. For both simulated and real datasets, we train our network with the AdamW optimizer using a constant learning rate of 0.001 for 150 epochs. The loss weight β_j is set to 1 for $j = 1 \ldots M-1$ and 2 for $j = M$. We evaluate our method's update rate on the test set, with a batch size of 1 to simulate sequential input image data, on the same computer (GPU: NVIDIA GeForce RTX 3090, CPU: AMD - Ryzen 5950 × 3.4 GHz 16-core 32-thread, RAM: 64 GB, OS: Ubuntu 20.04). Specifically, we measure the time between input data being passed to the network and shape sensing results being received. This includes not only the inference time but also data transfers between CPU and GPU, thus is more reflective of real-life update rate.

B. Quantitative Results

We report the shape sensing accuracy and runtime of our method, which are trained and tested on MoSS-Real and MoSS-Sim datasets separately. Additionally, MoSS-Real does not contain external effects besides gravity and friction from the sleeve. To explore if the method can generalize to robot shapes unseen during training, we collect an additional test set of 5 000 shapes on hardware with external forces, achieved by attaching a 20 g calibrated weight to the robot tip. The weight is attached with a blue polyethylene string to minimize the influence on the captured images, which results in an average shape disturbance of 56.9 mm and tip disturbance of 8.27 mm compared to the closest shapes seen in training. We refer to this test set as Disturbed-Real.

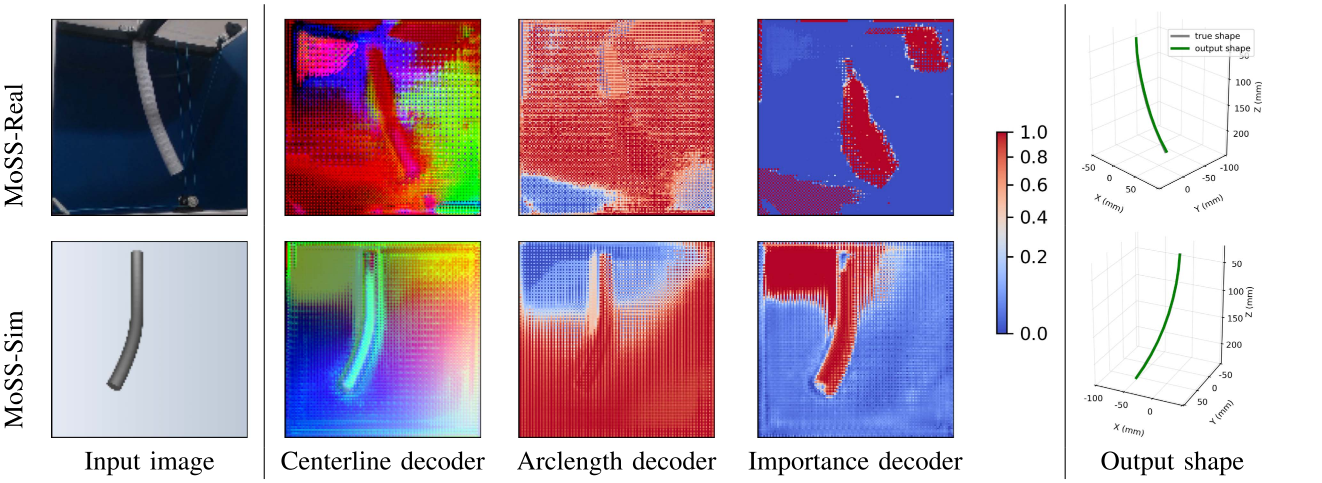


Fig. 4. MoSSNet's three decoders output interpretable pixel-wise information for 3D curve fitting despite not having pixel-wise supervision. The three encoder outputs, from left to right, provide spatial, length, and contour information about the robot to obtain accurate shape via weighted curve fitting.

TABLE II
Quantitative Performance of MoSSNet on MoSS-Sim, MoSS-Real, and an Additional Real Test Set Disturbed-Real With Unseen External Disturbance

Test set	MERS (mm) ↓	MERT (mm) ↓	fps ↑
MoSS-Sim	0.51 (0.20%)	0.88 (0.35%)	71.0
MoSS-Real	0.91 (0.36%)	1.85 (0.74%)	70.3
Disturbed-Real	1.98 (0.79%)	4.40 (1.76%)	72.0

Table II presents the metrics obtained from our evaluation. Specifically, MoSSNet achieves a mean error of robot shape (MERS) of 0.51 mm (0.20%) and a mean tip error (MERT) of $0.88\,\mathrm{mm}$ (0.35%) on the simulated test set. On the real test set, MoSSNet achieves a MERS of 0.91 mm (0.36%) and a MERT of 1.85 mm (0.74%). While our approach performs well on both datasets, it appears to perform slightly better on the simulated dataset due to the ideal conditions in that environment and the larger size of the dataset. The performance in the real-world setting is hindered slightly by the presence of noise in the captured images and sensor readings. Moreover, we observed that MoSSNet generalizes well on the disturbed test set, which is the real dataset with external forces that were not seen during training. The network achieves a MERS of 1.98 mm (0.79%) and a MERT of 4.40 mm (1.76%) mm on this test set. The model also achieves update rates over 70 fps consistently, which makes it suitable for real-time and even dynamic applications.

C. Qualitative Results

The network's performance can also be seen in Fig. 4, where test samples from MoSS-Real and MoSS-Sim are shown along with decoder outputs and shape output. All three decoder outputs provide interpretable pixel-wise information: 1) the centerline decoder predicts a 3D coordinate for each pixel in the image; 2) the arclength decoder predicts the pixel's relative position along the robot, and we observe increasing weight toward the bottom of the image as the robot points downward, with low weights at unreachable pixels; 3) the importance decoder outputs a nearly-binary segmentation of the robot, with noisy patches located at unreachable areas. The decoder outputs are then flattened and used for weighted linear least squares to obtain the robot shape. Compared to MoSS-Sim, the outputs from MoSS-Real are noisier due to more complex background and lighting condition, which aligns with our previous observation of higher shape sensing error on the real dataset.

The decoder outputs are noisy in general because only the ground truth shape is provided to the network during training, resulting in a lack of direct supervision on each pixel. This is a design decision made for practicality considerations, since per-pixel ground truth information is difficult to obtain on real continuum robots, especially if they are small in size. In our datasets, we include depth images to facilitate development in alternative approaches.

D. Sim-to-Real Transfer Learning

We conduct further experiments to evaluate transfer learning from the simulated to the real dataset. Specifically, we compare the performance of training the network from scratch to that of using a network pre-trained on the simulated dataset for the real test set. We run these experiments in various settings where the amount of available training data varied. Our results are presented in Fig. 5.

Generally, a larger amount of training data leads to lower test error. When only a small fraction of the real training dataset was available (e.g. 1% to 10%), pretraining the network on the simulated dataset leads to a significant improvement in performance on the real test set. However, when the amount of training data was large, pretraining on the simulated dataset does not provide additional benefit.

Thus, pre-training the network on simulation is beneficial when data collection on real hardware is not possible or costly. It is also worth noting that a model trained on simulated data only has very high MERS of 74.1 mm on MoSS-Real. This represent a significant sim-to-real gap caused by images in MoSS-Sim being very abstract and perceivably different from MoSS-Real. Having

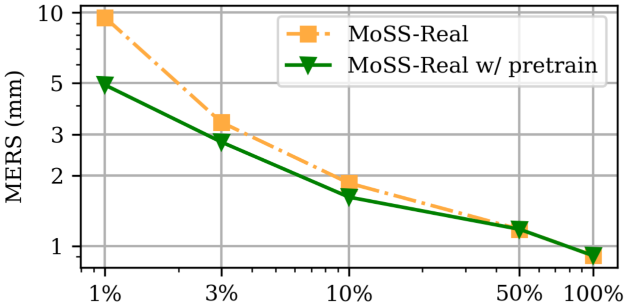


Fig. 5. Influence of the amount of real training data and pre-training with simulated data on MERS.

TABLE III
ABLATION STUDY OF THE PROPOSED COMPONENTS

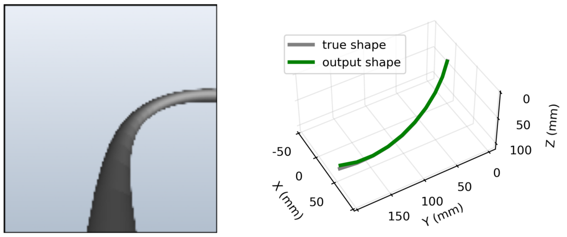
Decoders			Metrics (mm or fps)		
Centerline	Arclength	Importance	MERS ↓	MERT ↓	fps ↑
✓			1.12	2.22	72.2
✓	✓		0.99	1.93	71.6
✓	✓	✓	0.91	1.85	70.3

TABLE IV
ABLATION ANALYSIS ON POLYNOMIAL DEGREE

Polynomial Degree	MERS (mm)	MERT (mm)
2	2.06	2.22
3	0.96	**1.82**
4	**0.91**	1.85
5	1.01	2.22

TABLE V
QUANTITATIVE RESULTS ON MOSS-SIMWIDE

Test set	MERS (mm) ↓	MERT (mm) ↓	fps ↑
MoSS-SimWide	0.78 (0.32%)	1.61 (0.64%)	103



input image

output shape

Fig. 6. Qualitative results on MoSS-SimWide demonstrates the method is robust against changes in camera intrinsic and extrinsic parameters.

more realistic simulated images could improve the performance gain of pretraining.

E. Ablations

a) Influence of multiple decoders: We performed ablation studies to evaluate the contribution of each decoder, and the results are presented in Table III. In the first row, we use only one decoder to generate centerline coordinates and rely on the norm of the predicted 3D coordinates as a proxy for arclength during curve fitting. The addition of the arclength decoder enables us to predict the relative arclength accurately at each location within the captured image, resulting in a decrease of MERS by 0.13 mm and MERT by 0.29 mm. It's important to note that the arclength decoder has a more significant impact on reducing MERT, as it allows for better localization of the tip location. Furthermore, since not every pixel plays an equal role in regressing the robot shape, the introduction of the importance decoder assigns different weights for curve fitting to each pixel, leading to a further reduction in MERS to 0.91 mm and MERT to 1.85 mm.

b) Influence of polynomial degree for fitting: We performed an ablation analysis on the polynomial degree used for representing the robot shape. From the results summarized in Table IV, our chosen representation of degree 4 polynomials has the lowest MERS, while degree 3 yields very similar results and has a lower MERT. We think this design parameter is dependent on the robot – a more complex shape naturally requires higher-order representations, and that is why we see higher errors for degree 2 polynomial representation.

Interestingly, the errors increase for degree 5 polynomial, which indicates worse representation of the achievable robot shapes. It may also be beneficial to consider other basis functions for shape representation, such as Euler curves and Chebyshev polynomials; however, formal comparison between basis functions is outside the scope of this work.

F. Robustness to Camera Configuration

To verify that our method is robust to different imaging systems, we collected a new dataset on the simulation setup described in Section IV-B. The dataset contains 20 000 shapes captured in 512 × 512 resolution with the same 60% − 15%–25% split. The camera is placed 50 mm from the robot's base and has a view angle of 120 ° to simulate small wide-angle cameras used for endoscopy or industrial inspection, and we refer to this dataset as MoSS-SimWide.

Quantitative and qualitative results are shown in Table V and Fig. 6, respectively. The mean error of robot shape achieved is slightly higher compared to the MoSS-Sim dataset, which is possibly caused by the smaller dataset size and a higher level of robot self-occlusion. They demonstrate that our approach is robust to different camera intrinsic and extrinsic parameters and can potentially be applied to different imaging modalities depending on the specific application (e.g. laparoscopic imaging and X-ray imaging).

VI. LIMITATIONS AND FUTURE RESEARCH

While our method has shown good performance, the dataset collected in this work is not exhaustive in spanning all practical scenarios or environmental conditions (e.g., lighting, background, etc). Future work could improve robustness of the method to conditions outside the training dataset, or develop monocular methods on application-specific datasets, and quantitatively evaluate its robustness against lighting changes, imaging artifacts, and unseen configurations.

Furthermore, the method can be made more data efficient with sim-to-real domain randomization techniques. Self-supervision is also promising as unlabelled real data is inexpensive to collect in most cases.

Although the method only predicts the 3D position of the robot centerline, state estimation methods could be applied in tandem to obtain other robot states. The method could also be extended to incorporate temporal information for improved accuracy and robustness, and inspire future work in sensor fusion and shape control.

VII. Conclusion

We propose a novel monocular shape sensing method for continuum robots, called MoSSNet. Simulated and real datasets collected on a two-segment TDCR demonstrate the method is accurate (mean shape error of 0.91 mm((0.36%)), real-time (70 fps), and generalizes to unseen data. The method is also optimized end-to-end and does not require fiducial markers, segmentation, or camera calibration. MoSSNet outperforms existing stereo-vision-based shape sensing methods in terms of real-time capability and has much lower hardware complexity compared to embedded sensing methods. We believe that these promising results present a potential new alternative for continuum robot shape sensing. Additionly, we provide our code and dataset as part of the Open Continuum Robotics Project[1], serving as a research benchmarking tool.

Conflict of Interest

No competing financial interests exist.

Acknowledgment

The authors thank Raquel Urtasun for the insightful discussions related to Section III of this letter.

References

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

[2] M. Wang, X. Dong, W. Ba, A. Mohammad, D. Axinte, and A. Norton, "Design, modelling and validation of a novel extra slender continuum robot for in-situ inspection and repair in aeroengine," *Robot. Comput.-Integr. Manuf.*, vol. 67, 2021, Art. no. 102054.

[3] C. Kim, S. C. Ryu, and P. E. Dupont, "Real-time adaptive kinematic model estimation of concentric tube robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3214–3219.

[4] A. M. Franz, T. Haidegger, W. Birkfellner, K. Cleary, T. M. Peters, and L. Maier-Hein, "Electromagnetic tracking in medicine—A review of technology, validation, and applications," *IEEE Trans. Med. Imag.*, vol. 33, no. 8, pp. 1702–1725, Aug. 2014.

[5] C. Shi et al., "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 8, pp. 1665–1678, Aug. 2017.

[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 J.*, vol. 15, no. 11, pp. 6326–6334, Nov. 2015.

[7] S. Condino et al., "Electromagnetic navigation platform for endovascular surgery: How to develop sensorized catheters and guidewires," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 8, no. 3, pp. 300–310, 2012.

[8] F. Monet et al., "High-resolution optical fiber shape sensing of continuum robots: A comparative study," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8877–8883.

[9] Y. Cao, Z. Liu, H. Yu, W. Hong, and L. Xie, "Spatial shape sensing of a multisection continuum robot with integrated DTG sensor for maxillary sinus surgery," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 2, pp. 715–725, Apr. 2023.

[10] H. Donat, J. Gu, and J. J. Steil, "Real-time shape estimation for concentric tube continuum robots with a single force/torque sensor," *Front. Robot. AI*, vol. 8, 2021, Art. no. 734033.

[11] J. Burgner, S. D. Herrell, and R. J. Webster III, "Toward fluoroscopic shape reconstruction for control of steerable medical devices," in *Proc. ASME Dyn. Syst. Control Conf.*, 2011, pp. 791–794.

[12] C. Delmas et al., "Three-dimensional curvilinear device reconstruction from two fluoroscopic views," in *Proc. SPIE Med. Imag.: Image-Guided Procedures, Robotic Interv., Model.*, 2015, pp. 100–110.

[13] J. M. Croom, D. C. Rucker, J. M. Romano, and R. J. Webster, "Visual sensing of continuum robot shape using self-organizing maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 4591–4596.

[14] T. d. Veiga et al., "Challenges of continuum robots in clinical context: A review," *Prog. Biomed. Eng.*, vol. 2, no. 3, 2020, Art. no. 032003.

[15] 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 Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3130–3137, Apr. 2022.

[16] A. L. Orekhov, E. Z. Ahronovich, and N. Simaan, "Lie group formulation and sensitivity analysis for shape sensing of variable curvature continuum robots with general string encoder routing," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2308–2324, Jun. 2023.

[17] N. Nakagawa and H. Mochiyama, "Real-time shape estimation of an elastic rod using a robot manipulator equipped with a sense of force," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 8067–8073.

[18] J. Li, Y. Sun, H. Su, G. Zhang, and C. Shi, "Marker-based shape estimation of a continuum manipulator using binocular vision and its error compensation," in *Proc. IEEE Int. Conf. Mechatron. Automat.*, 2020, pp. 1745–1750.

[19] M. M. Dalvand, S. Nahavandi, and R. D. Howe, "High speed vision-based 3D reconstruction of continuum robots," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2016, pp. 000618–000623.

[20] D. B. Camarillo, K. E. Loewke, C. R. Carlson, and J. K. Salisbury, "Vision based 3-D shape sensing of flexible manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 2940–2947.

[21] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," 2018, *arXiv:1812.11941*.

[22] Y. Zhou, Y. He, H. Zhu, C. Wang, H. Li, and Q. Jiang, "Monocular 3D object detection: An extrinsic parameter free approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7556–7566.

[23] Y. Xu, J. Zhang, Q. Zhang, and D. Tao, "ViTPose++: Vision transformer for generic body pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–18, 2023.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[26] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, vol. 30, no. 1, 2013, pp. 1–6.

[27] W. Shi et al., "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1874–1883.

[28] D. C. Rucker and R. J. Webster III, "Statics and dynamics of continuum robots with general tendon routing and external loading," *IEEE Trans. Robot.*, vol. 27, no. 6, pp. 1033–1044, Dec. 2011.

[29] P. Rao, Q. Peyron, S. Lilge, and J. Burgner-Kahrs, "How to model tendon-driven continuum robots and benchmark modelling performance," *Front. Robot. AI*, vol. 7, 2021, Art. no. 630245.

[30] W. Schroeder, K. M. Martin, and W. E. Lorensen, *The Visualization Toolkit an Object-Oriented Approach to 3D Graphics*. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1998.

[1] [Online]. Available: https://www.opencontinuumrobotics.com