Shape Reconstruction of Soft, Continuum Robots using Differentiable Rendering with Geometrical Shape Primitive

Fei Liu[1], and Michael C. Yip[1]

***Abstract*— The shape information is essential for soft robots because it shows how the robot deforms during manipulation or navigation. Many previous works rely on embedded sensors, such as optical or electromagnetic, to capture the motion and localize in the 3D world. Alternatively, camera sensors are favorable because they are information-rich, easy to set up, and cost-effective. However, many existed vision-based works requires stereo images and depth data for reconstruction. In this work, we achieve image-based shape reconstruction of soft, continuum robot. Our method requires no precise robot meshes, but rather utilizes a differentiable renderer and a geometrical primitive, i.e., Bézier curve. It hence can be applied to soft robots even during deforming. Our parameter estimation pipeline is fully differentiable. The robot shape are estimated iteratively by back-propagating the image loss to update the parameters. We demonstrate that our method of using geometrical shape primitive can achieve high accuracy in shape reconstruction for a soft continuum robot.**

I. Introduction

Accurate sensory feedback of shape parameters is a cornerstone requirement for effectively operating autonomous systems in unfamiliar, real-world spaces. The drawback is that they all display cumulative errors as a result of deformable body configurations, in particular for soft, continuum robots.

The shape of the soft, continuum robot must be estimated when navigating in dynamic environments in order to increase the stability of the control system [1]. For soft robots, electromagnetic (EM) and Fiber Bragg Grating (FBG) sensors are integrated into the robot for measuring the state parameters directly [2]. A neural network may have been trained directly to map sensor readings to 3D shape such as in [3], [4]. However, the procedure of mounting the internal sensor can be cumbersome and extra efforts are required for designing the robot. Meanwhile, it requires significant amounts of labeled datasets which is usually infeasible for soft robots.

Alternatively, tracking and reconstruction the robot's shape configurations *directly from a camera* offers the greatest flexibility. They are easy to set up or are already recording, do not require access to internal robot sensors, are affordable and widely ubiquitous. Traditionally, fiducial markers, like ArUco marker [5] and AprilTag [6], are widely used for classical rigid robot pose estimation. These markers are attached to the specific locations of the robot and the robot state parameters can be estimated by knowing the kinematics model. However in most unstructured environments, it is unrealistic to have these markers attached. For soft robot applications, their body deformations and their tendency for full-body contact with environments and objects make it frequently impractical for securing fiducials or template markers.

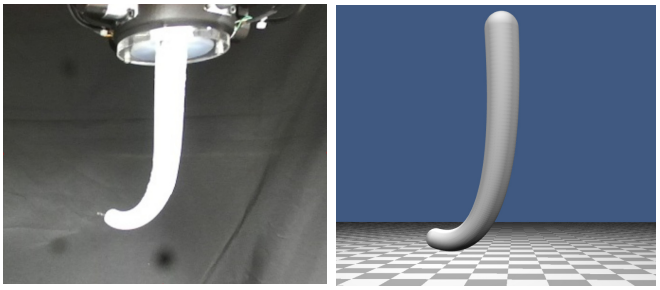


Fig. 1. Soft continuum robot shape reconstruction (right) with a Bézier geometrical primitive via differentiable rendering from real images (left).

In the literature, other imaged-based reconstruction works are very task- and environment-specific. Techniques include using fluoroscopy [7], [8] and ultrasound [9], [10]. These image-based techniques require specific imaging sources which might not always be available. [11], [12], [13] consider using endoscopic images for shape reconstruction while the markers are still required for identifying predefined feature points. Moreover, [14], [15] also introduce the shape reconstruction methods of using stereo images and depth data.

In contrast, we will be focusing on markerless shape reconstruction from a single RGB camera. Recently, in computer graphics, differentiable rendering has proved to be effective in image-based reconstruction by computing the derivative of images with respect to scene parameters such as object geometrical shape [16], [17], [18]. This could be translated to our task for the shape reconstruction.

In this paper, we demonstrate the capability of estimating 3D robot shape directly from a single image, as shown in Fig. 1. The method works via the technique of differentiable rendering, and can be effective for soft continuum robots. This is uniquely challenging, as soft continuum robots have an infinite number of configurations due to deformation. Under these constraints, we are still able to reconstruct the robot's shape through a predefined geometrical shape primitive, i.e., Bézier curve.

Our contributions are:

1) We propose a general framework for shape reconstruction by utilizing differentiable rendering with a single image.
2) The framework can be applied to any geometrical shape primitive; in this instance, we use a Bézier curve to estimate the parameters.

[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA.{f4liu, yip}@ucsd.edu

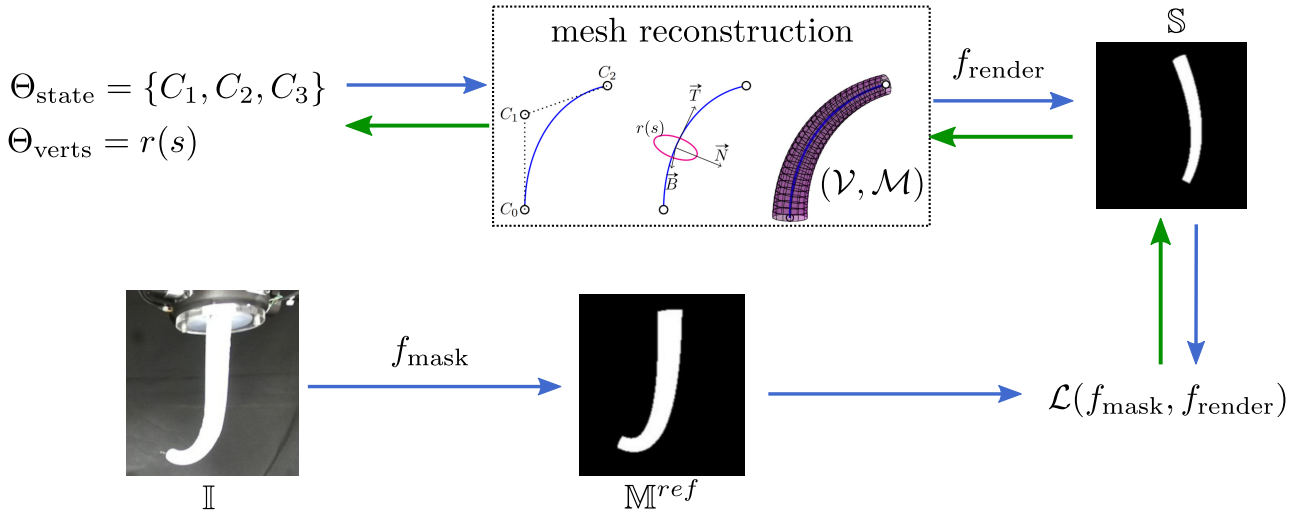


Fig. 2. The shape reconstruction framework of the soft continuum robot using differentiable rendering. The blue arrows are the forward pass, while the green arrows are the backward calculations.

3) We investigate the novel loss functions to overcome the local minima when applying differentiable rendering to the objective of robot pose estimation.

For a designed soft continuum robot, we collect real image datasets by deforming into different shape. We reconstruct the robot's shape by estimating the curve parameters to assess the effectiveness of our framework.

II. Methodology

We consider the problem of reconstruction the soft continuum robot shape parameters Θ from a single RGB image. Specifically, we estimate the robot shape by minimizing differences between the observed RGB image and a rendered reconstruction image. This is formulated as follows:

$$\Theta^* = \arg\min_{\Theta} \mathcal{L}(f_{mask}(\mathbb{I}), f_{render}(\Theta)) \tag{1}$$

where f_{mask} processed the given RGB image $\mathbb{I}$ into a binarized mask image for the robot. The function f_{render} takes in the estimated parameters, reconstructs the robot 3D shape, and renders the reconstruction. We aim to estimate the state parameters by minimizing the objective loss function $\mathcal{L}$. The process to get to this stage is described below.

A. The Shape Reconstruction Framework

The overall framework for state parameter estimation is illustrated in the Fig. 2. We first process the observed RGB image $\mathbb{I}$ into a binary mask $\mathbb{M}^{ref}$, which segments the robot pixel from the background. The binary mask contains value 1 for the pixels that belong to the robot and 0 otherwise. In our implementation, the segmentation is achieved by color segmentation for the soft continuum robot. We also initialize a robot mesh in a renderer as a set of geometrical primitive shapes with predefined vertices, edges, and faces.

During the iterative optimization process, we estimate the deformation of mesh vertices parameterized by Θ_{verts} and reconstruct the robot mesh with state parameters Θ_{state} (Section II-B). We render a silhouette image $\mathbb{S}$ from the reconstruction and compare it with the reference masked image $\mathbb{M}^{ref}$. A loss $\mathcal{L}$ is computed based on the curated objective functions (Section II-C). Since the full reconstruction and rendering pipeline is differentiable, we iterate the optimization process for N_o times and output the state parameters that minimize the objective loss with gradients-based method. The pipeline is detailed in Algorithm 1.

Algorithm 1: Robot Shape Reconstruction via Differentiable Rendering

Input : Image frame $\mathbb{I}$, initialization $\Theta^{(0)}_{state}, \Theta^{(0)}_{verts}$
Output: Reconstructed robot shape $\Theta^*_{state}, \Theta^*_{verts}$

// Generate robot masks
$\mathbb{M}^{ref} \leftarrow f_{mask}(\mathbb{I})$
// Optimization loop
$\mathcal{L}_{min} = \infty$
for $i = 0$ **to** N_o **do**
  // Section III-B
  $\mathcal{M}^{(i)} \leftarrow reconstructMesh(\Theta^{(i)}_{state}, \Theta^{(i)}_{verts})$
  // Section III-C
  $\mathbb{S}^{(i)} \leftarrow silhouetteRendering(\mathcal{M}^{(i)})$
  $\mathcal{L}^{(i)} \leftarrow computeLoss(\mathbb{S}^{(i)}, \mathbb{M}^{ref})$
  if $\mathcal{L}^{(i)} < \mathcal{L}_{min}$ **then**
    $\mathcal{L}_{min} = \mathcal{L}^{(i)}$
    $\Theta^*_{state} = \Theta^{(i)}_{state}$
  $\Theta^{(i+1)}_{verts} = \Theta^{(i)}_{verts} - \lambda_{verts} \frac{\partial \mathcal{L}^{(i)}}{\partial \Theta^{(i+1)}_{verts}}$
  $\Theta^{(i+1)}_{state} = \Theta^{(i)}_{state} - \lambda_{state} \frac{\partial \mathcal{L}^{(i)}}{\partial \Theta^{(i+1)}_{state}}$
return $\Theta^*_{state}, \Theta^*_{verts}$

B. Reconstruct Robot Mesh with Geometric Primitives

In this section, we describe the methods of reconstructing the robot mesh using geometric primitives for the soft robot. The shape of a soft continuum robot can be described in sev-

eral ways, most easily using a constant curvature model [19]. However, since this is a limiting approximation, instead a better model chosen is a Bézier curve model, which expresses a smooth and continuous curve with arbitrary curvature in 3D space. A Given a set of N control points $\{\mathbf{c}_i | \mathbf{c}_i \in \mathbb{R}^3\}_{i=0}^{N}$, the shape of the curve is defined as:

$$\mathbf{p}(s) = \sum_{i=0}^{N} \frac{N!}{i!\,(N-i)!}(1-s)^{N-i}s^i\mathbf{c}_i\ ,\ 0 \le s \le 1, \quad (2)$$

For simplicity, we use a quadratic Bézier curve ($N = 2$) and estimate the state of the control points Θ_{state} (see Fig. 2).

In general, the surface mesh for a continuum robot can be approximated as the tubular structure [20]. A tubular surface is defined as a union of cross sections, and each cross-section is centered at the axis along the 3D curve. To describe 3D coordinate frames along a quadratic Bézier curve, we compute the Frenet–Serret frame which is defined by a unit vector $\mathbf{T}$ tangent to the curve, a unit vector $\mathbf{N}$ normal to the curve, and a unit vector $\mathbf{B}$ perpendicular to the tangent and normal vectors. The Frenet–Serret coordinates, parameterized by s, are defined as:

$$\begin{aligned}
\mathbf{T}(s) &= \frac{\mathbf{p}'(s)}{\|\mathbf{p}'(s)\|} \\
\mathbf{N}(s) &= \frac{\mathbf{T}'(s)}{\|\mathbf{T}'(s)\|} = \frac{\mathbf{p}'(s) \times (\mathbf{p}''(s) \times \mathbf{p}'(s))}{\|\mathbf{p}'(s)\|\,\|\mathbf{p}''(s) \times \mathbf{p}'(s)\|} \\
\mathbf{B}(s) &= \mathbf{T}(s) \times \mathbf{N}(s) = \frac{\mathbf{p}'(s) \times \mathbf{p}''(s)}{\|\mathbf{p}'(s) \times \mathbf{p}''(s)\|}
\end{aligned} \quad (3)$$

where $\mathbf{p}'(s), \mathbf{p}''(s)$ are the first and second derivatives of the quadratic Bézier curve model:

$$\begin{aligned}
\mathbf{p}(s) &= (1-s)^2\mathbf{c}_0 + 2(1-s)s\mathbf{c}_1 + s^2\mathbf{c}_2 \\
\mathbf{p}'(s) &= 2(1-s)(\mathbf{c}_1 - \mathbf{c}_0) + 2s(\mathbf{c}_2 - \mathbf{c}_1) \\
\mathbf{p}''(s) &= 2(\mathbf{c}_2 - 2\mathbf{c}_1 + \mathbf{c}_0).
\end{aligned} \quad (4)$$

Each cross-section is approximated as a circle with the radius $\Theta_{verts} := r(s)$, and the corresponding tubular surface is defined as:

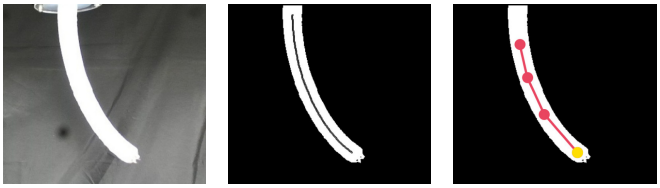
$$\mathbf{S}(s, \phi) = \mathbf{p}(s) + r(s)\,[-\mathbf{N}(s)\cos\phi + \mathbf{B}(s)\sin\phi] \quad (5)$$

with $\phi \in [0, 2\pi]$. Since a point on the tubular surface can be specified by s and ϕ, we compute the mesh vertices by discretizing the tubular surface. The mesh vertices are then defined by a set of points on the tubular surface with two additional points at both ends of the curve,

$$\mathcal{V} = \{\mathbf{S}(s_i, \phi_i), \mathbf{p}(0), \mathbf{p}(1) \mid i = 1, ..., N_d\} \quad (6)$$

where s_i, ϕ_i are discrete points for surface vertices.

The example of reconstructed surface mesh vertices $\mathcal{V}$ from the Bézier shape primitive are shown in Fig. 2. During the optimization process, we adjust these mesh vertices by optimizing the parameters Θ_{state} and Θ_{verts}. And finally, the adjusted vertices connections create the robot mesh $\mathcal{M}$.



(a) (b) (c)

Fig. 3. The pre-processing of soft octopus arm images, with the observed RGB image (left), the reference binary mask with center-line (middle) and the predefined keypoints and the endpoint (in yellow, right).

C. Differentiable Rendering

To render the image for robot mesh $\mathcal{M}$, we use the PyTorch3D differentiable render [21] for silhouette rendering. We set up the silhouette renderer with a perspective camera and a *SoftSilhouetteShader* which does not apply any lighting and shading. The differentiable renderer applies the rasterization algorithm which finds the mesh triangles that intersect each pixel and weights the influence according to the distance along the z-axis. Finally, the *SoftSilhouetteShader* computes pixel values of the rendered silhouette image using the sigmoid blending method [18].

D. Objective Loss Functions

To minimize the difference between the reconstructed silhouette image and the observed binary mask, the commonly used mask loss is applied. The mask loss computes the sum of the mean square error for every pixel,

$$\mathcal{L}_{mask} = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\left(\mathbb{S}(i,j) - \mathbb{M}^{ref}(i,j)\right)^2. \quad (7)$$

H and W is the image height and width, $\mathbb{S}$ is the rendered silhouette image and $\mathbb{M}^{ref}$ is the reference binary mask.

The mask loss will have non-informative gradients when there is no overlap (e.g. $\mathbb{S}(i,j) = 0$ but $\mathbb{M}^{ref}(i,j) = 1$). Therefore we use an additional keypoint loss to guide the optimization from local minima when the silhouettes do not overlap. The keypoints loss is defined as:

$$\mathcal{L}_{keypoint} = \sum_{i=1}^{K} \|\pi(\mathbf{p}_i) - \hat{\mathbf{x}}_i\|_2 \quad (8)$$

where K is the number of keypoints, $\hat{\mathbf{x}}_i$ is the i-th 2D keypoint extracted from center line of the reference mask $\mathbb{M}^{ref}$ as shown in Fig. 3, and $\mathbf{p}_i$ is the corresponding 3D keypoint on the Bézier curve. $\pi(\cdot)$ is the camera projection operator. Finally, the shape reconstruction loss is defined as:

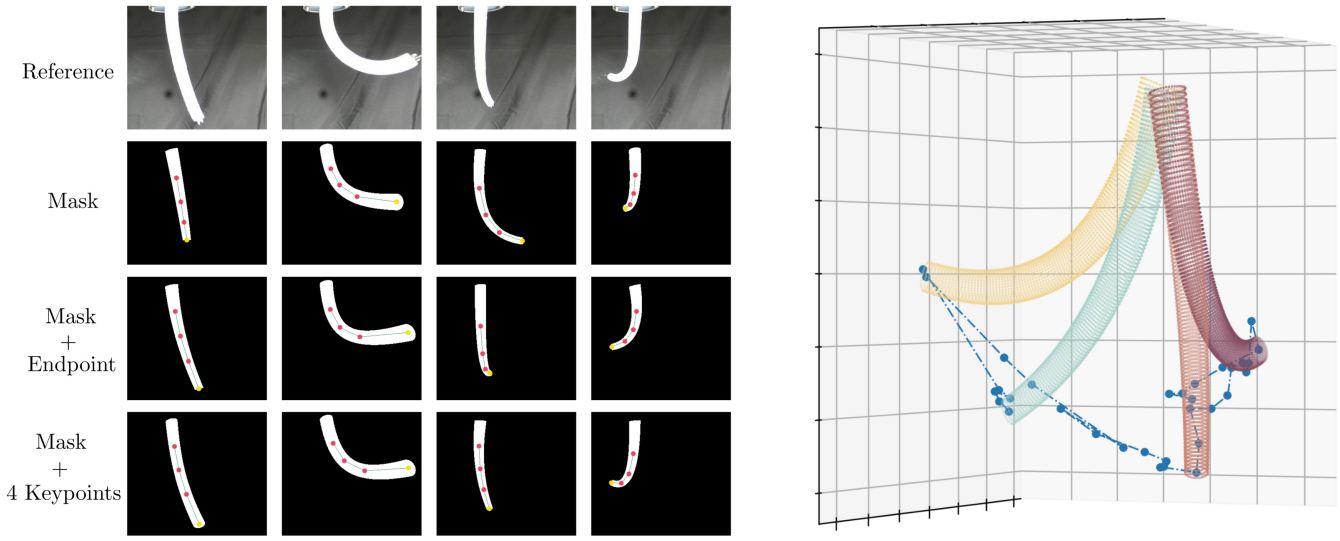
$$\mathcal{L}_{shape} = \lambda_{mask}\mathcal{L}_{mask} + \lambda_{keypoint}\mathcal{L}_{keypoint} \quad (9)$$

with $\lambda_{mask}, \lambda_{keypoint}$ as loss weights.

III. Experiments and Results

A. Datasets and Evaluation Metrics

Tendon-driven Octopus Arm dataset. Our experimental evaluations are conducted on a physical prototype of tendon-driven octopus arm, visible in Fig. 1. It consists of a tapered



(a) (b)

Fig. 4. Reconstruction results for the Octopus Arm dataset, with (a) the reference RGB images and shape reconstruction results for different losses are shown for 4 example frames that cover a large range of motion and (b) the reconstructed 3D robot shape of the picked frames and the entire robot trajectory.

cylinder of Ecoflex 00-30 (Smooth-On, Inc., Macungie, PA, USA) of length 200 mm, base and tip radius of 10 and 6 mm, respectively. It contains four channels for tendon actuation. The tendons are rigidly attached at the tip, and connected to spools actuated by harmonic drive motors at the base. The motors displace the tendons, leading to motions of the octopus arm. We collected the images using the ZED camera and the ground-truth robot shape is obtained with stereo reconstruction. For evaluation, we compare the center line of the reconstructed robot shape with the ground truth shape. The 2D and 3D errors of the center line are computed using the Euclidean distance of discrete points.

B. Shape Reconstruction for Soft Continuum Robot

1) Implementation details: The RGB images are pre-processed to binary masks and the 2D center-line are extracted from the reference binary mask using the *scikit-image* (`https://scikit-image.org`) package, which implements the fast skeletonization method [22]. We arbitrary predefined 4 keypoints along the center-line for loss computation, as shown in Fig. 3. For computing the mesh vertices, s is discretized to 100 and θ is discretized to 40 number of evenly spaced points. For the loss function, we set $\lambda_{mask} = 1$ and $\lambda_{keypoint} = 100$. We initialize the control points randomly but make sure the initialized mesh is within the camera frustum. The optimization loop is run for 200 iterations with a learning rate of 0.2.

TABLE I

2D AND 3D ERROR (MEAN e AND STANDARD DEVIATION σ) OF SHAPE RECONSTRUCTION ON REAL OCTOPUS ARM DATASET.

Losses	e_{2D} (pixel)	σ_{2D}	e_{3D}	σ_{3D}
Mask	12.462	8.690	8.720	2.534
Mask + endpoint	3.898	2.216	7.299	3.900
Mask + 4 keypoints	**3.276**	**0.785**	**6.915**	**2.096**

2) Evaluation on the Octopus Arm Dataset: We evaluate our shape reconstruction method with different loss functions described in Section II-D. For the keypoint loss, we experimented with only using the endpoint and using all 4 keypoints. We report the averaged 2D and 3D center-line error and the results are shown in Table I. We can see that the error is dropped significantly by combining the mask loss and keypoint loss with only the endpoint. Considering more keypoints further improves our performance of shape reconstruction as they provide more guidance for optimization. The qualitative results are shown in the Fig. 4, where we show the rendered silhouette images of the reconstructed robot mesh (left) and, the reconstructed 3D robot shape, and the robot trajectory (right).

IV. CONCLUSION

In this paper, we demonstrate the capability of reconstruction the soft's shape directly from a camera, as shown in Fig. 1. The method works via the technique of differentiable rendering, and can be generalized to any geometrical shape primitives. We show that several definitions for optimization losses are useful to overcome the local minima when applying differentiable rendering to the objective functions. We evaluated our method on relatively unstructured environments of continuum robot showing its efficacy. Ultimately, this work helps to enable robot shape reconstruction and tracking without embedded sensors, with greater opportunities in useful dataset curation, behavioral cloning and visual learning. For future work, we will incorporate the shape reconstruction for close-loop robot manipulation tasks.

REFERENCES

[1] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "A soft, steerable continuum robot that grows via tip extension," *Soft Robotics*, vol. 6, no. 1, pp. 95–108, 2019. PMID: 30339050.

[2] 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, pp. 1665–1678, 2016.

[3] Q. Zhao, J. Lai, K. Huang, X. Hu, and H. K. Chu, "Shape estimation and control of a soft continuum robot under external payloads," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 2511–2522, 2022.

[4] X. T. Ha, D. Wu, M. Ourak, G. Borghesan, J. Dankelman, A. Menciassi, and E. V. Poorten, "Shape sensing of flexible robots based on deep learning," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 1580–1593, 2023.

[5] S. Garrido-Jurado *et al.*, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[6] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE international conference on robotics and automation*, pp. 3400–3407, IEEE, 2011.

[7] C. Papalazarou, P. M. Rongen, *et al.*, "3d catheter reconstruction using non-rigid structure-from-motion and robotics modeling," in *Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 8316, pp. 622–629, SPIE, 2012.

[8] M. Hoffmann, A. Brost, C. Jakob, F. Bourier, M. Koch, K. Kurzidim, J. Hornegger, and N. Strobel, "Semi-automatic catheter reconstruction from two views," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 584–591, Springer, 2012.

[9] M. Waine, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, "3d shape visualization of curved needles in tissue from 2d ultrasound images using ransac," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4723–4728, IEEE, 2015.

[10] S. Cheung and R. Rohling, "Enhancement of needle visibility in ultrasound-guided percutaneous procedures," *Ultrasound in medicine & biology*, vol. 30, no. 5, pp. 617–624, 2004.

[11] A. AlBeladi, G. Krishnan, M.-A. Belabbas, and S. Hutchinson, "Vision-based shape reconstruction of soft continuum arms using a geometric strain parametrization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11753–11759, 2021.

[12] P. Cabras, F. Nageotte, P. Zanne, and C. Doignon, "An adaptive and fully automatic method for estimating the 3d position of bendable instruments using endoscopic images," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 4, p. e1812, 2017. e1812 RCS-16-0177.R2.

[13] R. Reilink, S. Stramigioli, and S. Misra, "Pose reconstruction of flexible instruments from endoscopic images using markers," in *2012 IEEE International Conference on Robotics and Automation*, pp. 2938–2943, 2012.

[14] J. M. Croom, D. C. Rucker, J. M. Romano, and R. J. Webster, "Visual sensing of continuum robot shape using self-organizing maps," in *2010 IEEE International Conference on Robotics and Automation*, pp. 4591–4596, 2010.

[15] S. Xu, G. Li, D. Song, L. Sun, and J. Liu, "Real-time shape recognition of a deformable link by using self-organizing map," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 586–591, 2018.

[16] H. Kato, Y. Ushiku, and T. Harada, "Neural 3d mesh renderer," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3907–3916, 2018.

[17] H. Kato, D. Beker, M. Morariu, T. Ando, T. Matsuoka, W. Kehl, and A. Gaidon, "Differentiable rendering: A survey," *arXiv preprint arXiv:2006.12057*, 2020.

[18] S. Liu, T. Li, W. Chen, and H. Li, "Soft rasterizer: A differentiable renderer for image-based 3d reasoning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7708–7717, 2019.

[19] I. Robert J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[20] S. Li and G. Hao, "Current trends and prospects in compliant continuum robots: A survey," *Actuators*, vol. 10, no. 7, 2021.

[21] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv preprint arXiv:2007.08501*, 2020.

[22] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, no. 3, pp. 236–239, 1984.