2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Image-based Pose Estimation and Shape Reconstruction for Robot Manipulators and Soft, Continuum Robots via Differentiable Rendering

Jingpei Lu[1,†], Fei Liu[1,†], Cédric Girerd[1,2], and Michael C. Yip[1]

***Abstract*— State estimation from measured data is crucial for robotic applications as autonomous systems rely on sensors to capture the motion and localize in the 3D world. Among sensors that are designed for measuring a robot's pose, or for soft robots, their shape, vision sensors are favorable because they are information-rich, easy to set up, and cost-effective. With recent advancements in computer vision, deep learning-based methods no longer require markers for identifying feature points on the robot. However, learning-based methods are data-hungry and hence not suitable for soft and prototyping robots, as building such bench-marking datasets is usually infeasible. In this work, we achieve image-based robot pose estimation and shape reconstruction from camera images. Our method requires no precise robot meshes, but rather utilizes a differentiable renderer and primitive shapes. It hence can be applied to robots for which CAD models might not be available or are crude. Our parameter estimation pipeline is fully differentiable. The robot shape and pose are estimated iteratively by back-propagating the image loss to update the parameters. We demonstrate that our method of using geometrical shape primitives can achieve high accuracy in shape reconstruction for a soft continuum robot and pose estimation for a robot manipulator.**

I. Introduction

Sensory feedback of state parameters, such as the position and body configuration of a robot in its environment, is a fundamental requirement for operating autonomous systems in real-world, unknown spaces. In-place sensing may exist with motor encoders for robot manipulators, or Fiber Braggs for soft robots [1], all providing an estimate of their relative pose and body configurations in relation to the real world environment; however, the limitation is they all exhibit cumulative position errors due to long kinematic chains. In addition, for both soft and rigid robots, the procedure for mounting internal sensors can be tricky and wiring and communication lines can constrain the mechanics and articulation of the robot. That is why measuring the body configuration of a soft robot is notoriously challenging.

If the goal is to observe and track the motion of robots in the wild, e.g., for behavior cloning or offline reinforcement learning, the robot's state information may not be readily available or even accessible. A good example is in minimally invasive surgery (MIS), where over 1 million procedures are performed yearly on a daVinci Surgical Robot, many researchers are interested in automating aspects of the surgical procedures [2]. Video of surgeons all over the world using the same robot to perform tasks like suturing, where properly grasping suture needles is an expert skill for which data could be useful for learning control policies as shown in [3]. However, these datasets only have video data and lack kinematic data from the robot due to proprietary access.

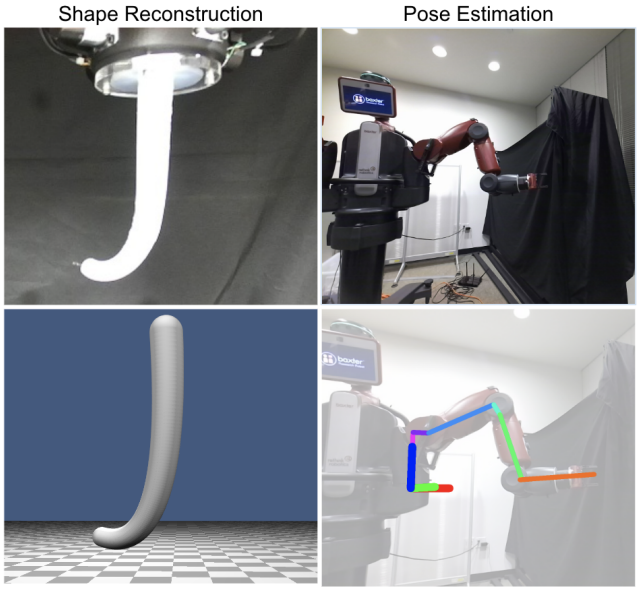


Fig. 1. Robot shape reconstruction and pose estimation via differentiable rendering. The top row shows the real images. The bottom row shows the estimated robot shape (left) and pose (right).

Ultimately, in the above MIS and many other *in-the-wild* robot scenarios, tracking the robot pose and body 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 [4] and AprilTag [5], are widely used for 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; furthermore, in dirty environments like MIS or constrained environments, these markers can be permanently obscured or occluded. 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.

The most flexible way to estimate pose from a camera is to do marker-less tracking. With recent advancements in Com-

This project was funded by the US Army Telemedicine and Advanced Technologies Research Center and NSF Awards #2045803 and #1935329.

† These authors contributed equally.

[1]Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA.{jil360, f4liu, cgirerd, yip}@ucsd.edu

[2]LIRMM, Univ Montpellier, CNRS, Montpellier, France. cedric.girerd@lirmm.fr. This work was conducted while he was a visiting scholar at UCSD.

puter Vision, Convolutional Neural Networks (CNN) present a promising way for marker-less feature detection [6][7] which no longer requires physical modification on the robot. In spite of the success of the CNN, training a neural network requires significant amounts of labeled datasets which is usually infeasible for soft robots and other robot prototypes, or where labeling of data is costly (e.g., MIS). 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 camera pose and object geometry [8][9][10]. This could be translated to the task of marker-less pose tracking.

In this paper, we demonstrate the capability of estimating robot pose and configuration directly from a camera, as shown in Fig. 1. The method works via the technique of differentiable rendering, and can be effective both in rigid-link robot manipulators as well as soft continuum robots. This is uniquely challenging, as soft continuum robots have an infinite number of configurations, while some rigid robot manipulators may not come with predefined CAD models for their users (these are typically proprietary). Under these constraints, we are still able to estimate a robot's state without knowledge of a high-resolution CAD by introducing a flexible method involving shape primitives that work across a wide range of robots. Our contributions are:

1) We propose a general framework for parameter estimation by utilizing differentiable rendering with geometrical shape primitives.
2) The framework generalizes to both rigid and soft continuum robots for parameter estimation.
3) We investigate the novel loss functions to overcome the local minima when applying differentiable rendering to the objective of robot pose estimation.

To examine the effectiveness of our framework, we collect a synthetic and a real dataset for a soft continuum robot and reconstruct the robot shape by estimating the curve parameters. We also evaluate our method on robot pose estimation where the 6 Degree-of-Freedom (DOF) camera-to-robot pose for a Baxter robot is estimated with provided RGB images and joint encoder readings. The experimental results show that our method is out-performing the state-of-the-art pose estimation algorithms.

II. Previous Work

A few techniques in image-based estimation of robot poses and shapes have been previously explored.

Image-based Shape Reconstruction for Soft Continuum Robots: Image-based measurements of soft continuum robots are very task-specific and only work in certain environments. Techniques include using fluoroscopy [11], [12] and ultrasound [13], [14]. These image-based techniques require specific imaging sources which might not always be available. [15], [16], [17] consider using endoscopic images for shape reconstruction while the markers are required for identifying predefined feature points. Moreover, [18], [19] 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.

Image-based Robot Pose Estimation: The common approach for image-based robot pose estimation is to attach the fiducial markers [4], [5] to known locations along the robot kinematic chain. Given the joint angles, the position of the marker in the robot base frame can be calculated and the robot pose can be derived by solving an optimization problem [20], [21], [22]. More recently, deep learning brings a promising way of marker-less pose estimation, where a CNN is trained to extract predefined feature points and the robot pose is estimated by solving the Perspective-n-Point problem [23], [6], [24], [7]. Meanwhile, rendering-based methods also demonstrate their advantages in robot pose estimation by using the high-resolution robot CAD model for more precise estimation [25], [26]. Our work utilizes differentiable rendering which requires no robot CAD model or large dataset for model training.

III. Methodology

We consider the problem of estimating the robot state parameters Θ from a single RGB image. Specifically, we estimate the robot pose and configurations 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 mesh, and renders the reconstruction. We aim to estimate the state parameters by minimizing the objective loss function $\mathcal{L}$. A visual of the final optimization process are shown in Fig. 2. The process to get to this stage is described below.

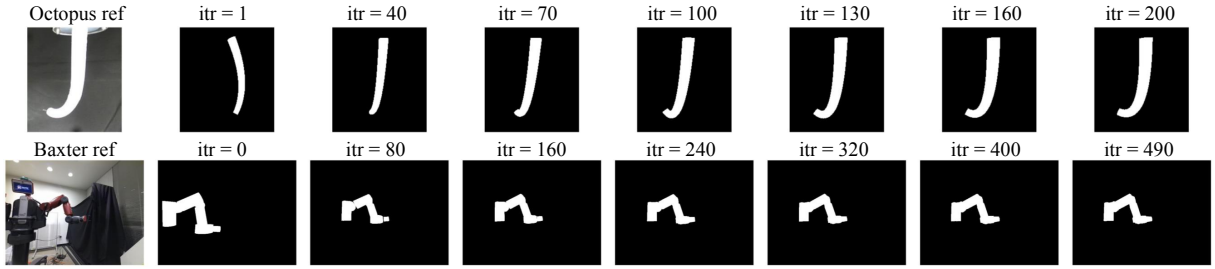


Fig. 2. Visualization of the optimization process. We show the rendered silhouettes at different iterations to demonstrate the convergence of our algorithm.

Algorithm 1: Robot State Parameter Estimation via Differentiable Rendering

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

// Generate robot masks
1 $\mathbb{M}^{ref} \leftarrow f_{mask}(\mathbb{I})$
// Optimization loop
2 $\mathcal{L}_{min} = \infty$
3 **for** $i = 0$ **to** N_o **do**
// Section III-B
4 $\mathcal{M}^{(i)} \leftarrow reconstructMesh(\Theta^{(i)}_{state}, \Theta^{(i)}_{verts})$
// Section III-C
5 $\mathbb{S}^{(i)} \leftarrow silhouetteRendering(\mathcal{M}^{(i)})$
6 $\mathcal{L}^{(i)} \leftarrow computeLoss(\mathbb{S}^{(i)}, \mathbb{M}^{ref})$
7 **if** $\mathcal{L}^{(i)} < \mathcal{L}_{min}$ **then**
8 $\mathcal{L}_{min} = \mathcal{L}^{(i)}$
9 $\Theta^{*}_{state} = \Theta^{(i)}_{state}$
10 $\Theta^{(i+1)}_{verts} = \Theta^{(i)}_{verts} - \lambda_{verts} \frac{\partial \mathcal{L}^{(i)}}{\partial \Theta^{(i+1)}_{verts}}$
11 $\Theta^{(i+1)}_{state} = \Theta^{(i)}_{state} - \lambda_{state} \frac{\partial \mathcal{L}^{(i)}}{\partial \Theta^{(i+1)}_{state}}$
12 **return** Θ^{*}_{state}

A. The State Parameter Estimation Framework

The overall framework for state parameter estimation is described in the Algorithm 1. 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 (Section IV-B) and a CNN-based semantic segmentation for the robot manipulator (Section IV-C). 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 III-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 III-C). Since the full reconstruction and rendering pipeline is differentiable, a gradient on the loss may be taken with respect to the parameters and the objectives can be optimized (lines 11-12 in Algorithm 1). We iterate the optimization process for N_o times and output the state parameters that minimize the objective loss.

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 and rigid robot, respectively. Note that state parameters Θ_{state} and mesh vertex parameters Θ_{verts} are defined differently according to their body types.

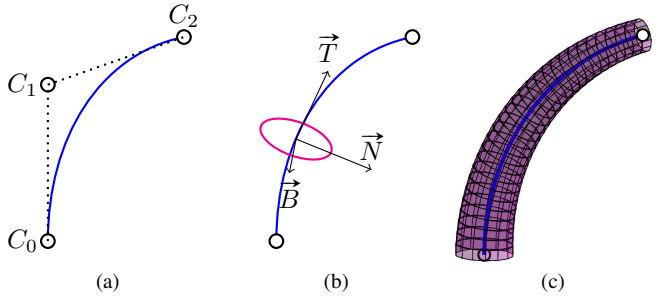


Fig. 3. Illustration of surface mesh construction for soft continuum robot, with (a) the Bézier curve and control points, (b) the Frenet–Serret frame of cross section, and (c) the constructed surface mesh.

1) Mesh Reconstruction for Soft Continuum Robot: The shape of a soft continuum robot can be described in several ways, most easily using a constant curvature model [27]. 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. 3a).

In general, the surface mesh for a continuum robot can be approximated as the tubular structure [28]. A tubular surface is defined as a union of cross sections, and each cross-section is centered at the axis along the 3D curve, as shown in Fig 3. 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 (Fig. 3b). 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 $r(s)$, and the corresponding tubular surface is defined as:

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

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,

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

where s_i, ϕ_i are discrete points for surface vertices. The example of reconstructed surface mesh is shown in Fig. 3c. During the optimization process, we adjust the mesh vertices by optimizing the radius of the cross sections $\Theta_{verts} := r(s)$, and the robot mesh is formed with adjusted vertices.

2) Mesh Reconstruction for Robot Manipulator: A robot manipulator can generally be described as a chain of rigid links, and the 3D robot mesh is separated into a set of individual meshes of primitive shape. The number of individual meshes equals the number of rigid links and the meshes are connected by the rigid body transformations which indicate the position and rotation with respect to the robot base frame $\{b\}$. The transformation matrices $\mathbf{T}_n^b(q_1, ..., q_n) \in SE(3)$ are parameterized by joint angles and can be computed from forward kinematics [29].

We initialize the set of individual meshes as primitive shapes (e.g. boxes or cylinders) with predefined edges, faces, and vertices $\mathcal{V}_{\texttt{primitive}}$. Note that anything can be a primitive shape as long as there are only a few tunable parameters defining it, so a link shape template could be used. Regardless, during the optimization process, we adjust the robot mesh by estimating the offsets of each vertex:

$$\mathbf{v} = \mathbf{v}_{\texttt{primitive}} + \mathbf{v}_{\texttt{offset}} \tag{7}$$

where $\mathbf{v}_{\texttt{primitive}} \in \mathcal{V}_{\texttt{primitive}}$ is the vertex initialized with the primitive shape mesh and $\mathbf{v}_{\texttt{offset}}$ is the corresponding offset vector. The set of offset vectors has the same number of elements as $\mathcal{V}_{\texttt{primitive}}$, and is optimized at each iteration through gradient descent ($\Theta_{verts} := \mathcal{V}_{\texttt{offset}}$).

To compose the robot mesh $\mathcal{M}$ for pose estimation, each individual mesh needs to be connected by the forward kinematics and transformed to the camera frame. Let $\mathbf{v}^n \in \mathbb{R}^3$ be a vertex of the n-th link mesh, we transform the vertex to the camera frame as:

$$\overline{\mathbf{v}}^c = \mathbf{T}_b^c \mathbf{T}_n^b(q_1, ..., q_n) \overline{\mathbf{v}}^n \tag{8}$$

where $\bar{\cdot}$ represents the homogeneous representation of a point (e.g. $\overline{\mathbf{v}} = [\mathbf{v}, 1]^T$). $\mathbf{T}_n^b(q_1, ..., q_n)$ obtained from forward kinematics transforms mesh vertices to the robot base frame and $\mathbf{T}_b^c$ is the robot-to-camera transformation which is estimated using the Algorithm 1 ($\Theta_{state} := \mathbf{T}_b^c$).

C. Differentiable Rendering and Loss Functions

To render the image for robot mesh $\mathcal{M}$, we use the PyTorch3D differentiable render [30] 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 [10].

1) Objective Loss Functions for Shape Reconstruction: 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,

$$\mathcal{L}_{mask} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \left(\mathbb{S}(i,j) - \mathbb{M}^{ref}(i,j)\right)^2. \tag{9}$$

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:

$$\mathcal{L}_{keypoint} = \sum_{i=1}^{K} \|\pi(\mathbf{p}_i) - \hat{\mathbf{x}}_i\|_2 \tag{10}$$

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. 5, 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:

$$\mathcal{L}_{shape} = \lambda_{mask}\mathcal{L}_{mask} + \lambda_{keypoint}\mathcal{L}_{keypoint} \tag{11}$$

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

2) Objective Loss Functions for Pose Estimation: For robot pose estimation, poor initialization would hamper the performance of the optimization algorithm by converging to local minima. In addition to the commonly used mask loss (Eq. 9), we propose distance loss and appearance loss to aid the optimization. The distance loss utilizes the distance map to propagate the gradient information to the entire image. The distance map $\mathbb{D}^{ref}$ is defined as:

$$\mathbb{D}^{ref}(i,j) = \begin{cases} 0, & \text{if } \mathbb{M}^{ref}(i,j) = 1 \\ \frac{dist(i,j)}{\gamma}, & \text{otherwise} \end{cases} \tag{12}$$

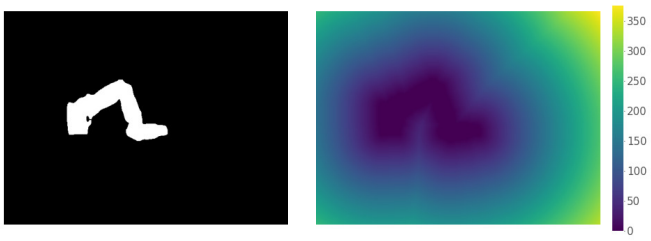
where $dist(i,j)$ is the distance from the pixel (i,j) to the closest pixel that has positive value 1, and γ is a discount factor (Fig. 4). We use the *scikit-fmm* package* which implements the fast marching method [31] for computing the distance map. The distance loss is then calculated as:

$$\mathcal{L}_{dist} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \mathbb{S}(i,j) * \mathbb{D}^{ref}(i,j) \tag{13}$$

Since the reconstructed mask should have the same appearance as the observed mask, we introduce the appearance loss to force them to have the same number of positive pixels:

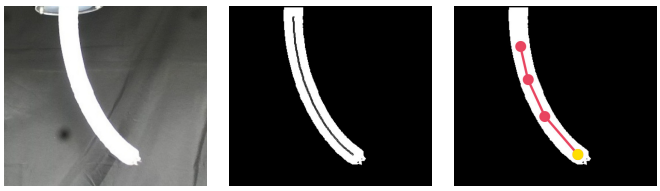
$$\mathcal{L}_{app} = \left\| \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \mathbb{S}(i,j) - \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \mathbb{M}^{ref}(i,j) \right\| \tag{14}$$

*https://pythonhosted.org/scikit-fmm



(a) (b)

Fig. 4. Visualization of the distance map, with (a) the reference binary mask and (b) the corresponding distance map.



(a) (b) (c)

Fig. 5. The pre-processing of soft octopus arm images, with (a) the observed RGB image, (b) the reference binary mask with center-line and (c) the predefined keypoints and the endpoint (yellow).

The appearance loss is effective for preventing the robot pose from being too far or too close to the camera, through regulating the size of the rendered mask. Finally, the objective loss for pose estimation is defined as:

$$\mathcal{L}_{pose} = \lambda_{mask}\mathcal{L}_{mask} + \lambda_{dist}\mathcal{L}_{dist} + \lambda_{app}\mathcal{L}_{app} \quad (15)$$

where $\lambda_{mask}, \lambda_{dist}, \lambda_{app}$ are weights for the loss functions.

IV. 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 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.

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**

Baxter dataset. The Baxter dataset from [7] provides 100 image frames of 20 different robot poses. The ground-truth end-effector position in 2D and 3D are provided. This dataset includes the challenging scenarios where the robot manipulator is self-occluded. The Percentage of Correct Keypoints (PCK) metric of the end-effector will be calculated according to [7], where the end-effector position in the camera frame is calculated based on the estimated robot pose.

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 [32]. We arbitrary predefined 4 keypoints along the center-line for loss computation, as shown in Fig. 5. 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.

2) Evaluation on the Octopus Arm Dataset: We evaluate our shape reconstruction method with different loss functions described in Section III-C.1. 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. 6, where we show the rendered silhouette images of the reconstructed robot mesh (left) and, the reconstructed 3D robot shape, and the robot trajectory (right).

C. Pose Estimation for Robot Manipulator

1) Implementation details: To segment the robot from the background, we trained the DeepLabV3 [33] with 10K synthetic image data generated using the CoppeliaSim [34]. We applied Domain Randomization [35] so that the trained network can generalize to the real images. We initialize the primitive shape meshes for each link with the length, width, and height that are described in the robot description file. The deformed vertices $\mathbf{v}_{\texttt{deformed}}$ are initialized to zeros and the robot poses are initialized randomly but within the camera frustum. For loss function, we set the weights as $\lambda_{mask} = 1, \lambda_{dist} = 1, \lambda_{app} = 1$ and $\gamma = 100$ when computing the distance map. We optimize parameters for 500 iterations with a learning rate of 1e-2 for camera pose parameters and 1e-4 for the deformed vertices.

2) Evaluation on Baxter Dataset: We evaluate our method of robot pose estimation on the Baxter dataset and compare it against the state-of-the-art methods [6], [7], as shown in

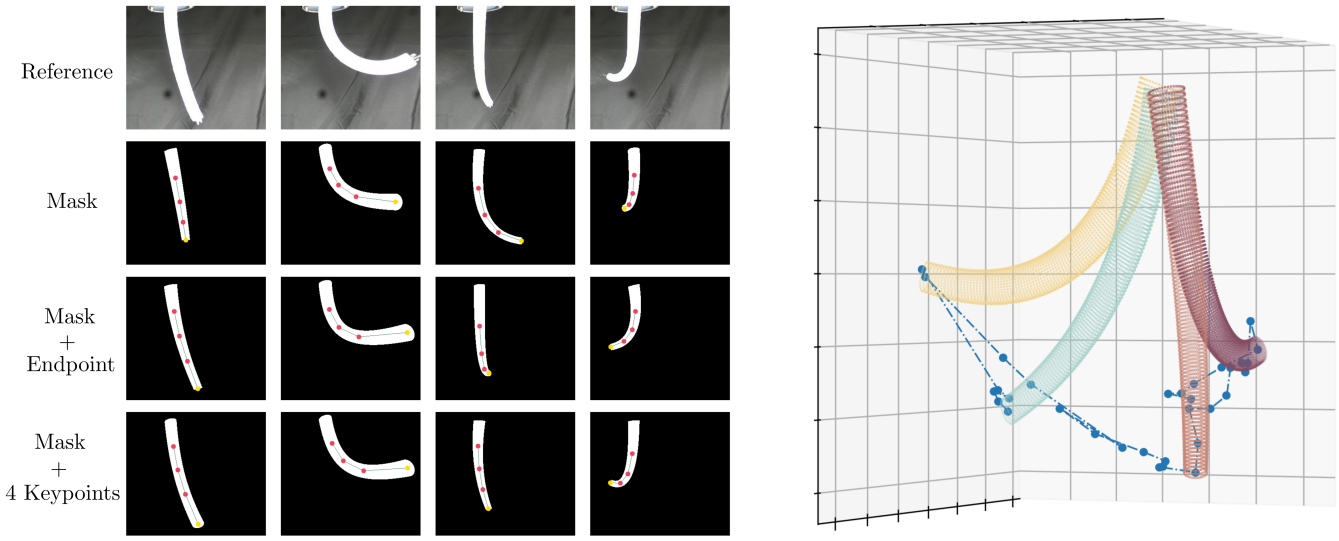


Fig. 6. Reconstruction results for the Octopus Arm dataset, with (a) the reference RGB images and shape reconstruction results for different losses are shown for 4 example frames that cover a large range of motion and (b) the reconstructed 3D robot shape of the picked frames and the entire robot trajectory.

TABLE II

COMPARISON OF OUR METHODS WITH THE STATE-OF-THE-ART METHODS ON ROBOT POSE ESTIMATION.

	PCK2D				PCK3D			
	@50 pixel	@100 pixel	@150 pixel	@200 pixel	@100 mm	@200 mm	@300 mm	@400 mm
DREAM [6]	0.33	0.52	0.62	0.64	0.32	0.43	0.54	0.66
Optimized Keypoints [7]	0.69	0.88	0.93	0.95	0.47	0.74	0.86	0.90
Ours (box)	0.65	**0.94**	**0.95**	0.95	**0.8**	**0.95**	0.95	0.95
Ours (cylinder)	**0.80**	0.91	0.93	0.95	0.71	0.93	0.94	0.95
Ours (CAD)	0.74	0.90	0.94	**1.0**	0.78	0.93	**0.97**	**1.0**

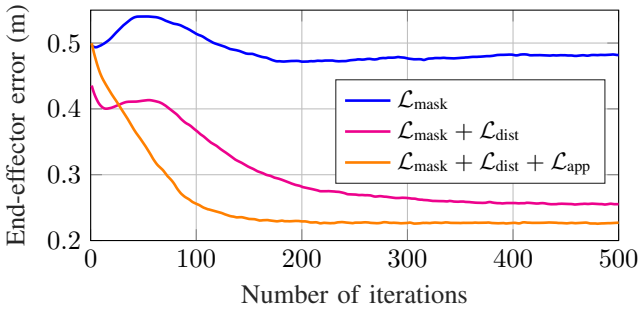


Fig. 7. Ablation study of the loss functions for robot pose estimation. Notice that the averaged error is large because of outliers, and 95% of the estimates have less than 4 cm error as shown in Table II.

Table II. We applied our method with two different shape primitive meshes, the box and cylinder. We also report the performance of using the robot CAD model with differentiable rendering. The Percentage of Correct Keypoints (PCK, higher is better) results are reported for both 2D and 3D at different thresholds. The experimental results show that our method of using primitive shapes outperforms the state-of-the-art methods and achieves comparable performance with using the high-resolution robot CAD model.

3) Ablation Study on Loss Functions: Here, we study the effectiveness of the loss functions proposed in Section III-C.2 using the Baxter dataset. We experiment with our robot pose estimation method (cylinder) with different loss combinations and calculate 3D end-effector error using the Euclidean Distance. We plot the average 3D end-effector error at each iteration in Fig. 7. With only the mask loss, the algorithm suffers from bad initialization and cannot converge robustly. The distance loss helps the convergence by propagating the gradient information to every image pixel. Finally, by combining all three losses, we achieve a more robust convergence for robot pose estimation.

V. CONCLUSION

In this paper, we demonstrate the capability of measuring robot pose and configuration state parameters directly from a camera, as shown in Fig. 1. The method works via the technique of differentiable rendering, and can be effective both in rigid-link robot manipulators as well as soft continuum robots. we show that several definitions for optimization losses are useful to overcome the local minima when applying differentiable rendering to the objective of robot pose estimation. We evaluated our method on relatively unstructured environments of continuum and rigid robots showing its efficacy in pose estimation. Ultimately, this work helps to enable robot state estimation and tracking *in the wild*, with greater opportunities in useful dataset curation, behavioral cloning and visual learning.

REFERENCES

[1] 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.

[2] M. Yip and N. Das, "Robot autonomy for surgery," in *The Encyclopedia of MEDICAL ROBOTICS: Volume 1 Minimally Invasive Surgical Robotics*, pp. 281–313, World Scientific, 2019.

[3] Z.-Y. Chiu, F. Richter, E. K. Funk, R. K. Orosco, and M. C. Yip, "Bimanual regrasping for suture needles using reinforcement learning for rapid motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7737–7743, IEEE, 2021.

[4] 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.

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

[6] T. E. Lee, J. Tremblay, T. To, J. Cheng, T. Mosier, O. Kroemer, D. Fox, and S. Birchfield, "Camera-to-robot pose estimation from a single image," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9426–9432, IEEE, 2020.

[7] J. Lu, F. Richter, and M. C. Yip, "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4622–4629, 2022.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] Y. Li, F. Richter, J. Lu, E. K. Funk, R. K. Orosco, J. Zhu, and M. C. Yip, "Super: A surgical perception framework for endoscopic tissue manipulation with surgical robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2294–2301, 2020.

[21] J. Ilonen and V. Kyrki, "Robust robot-camera calibration," in *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 67–74, IEEE, 2011.

[22] F. Richter, J. Lu, R. K. Orosco, and M. C. Yip, "Robotic tool tracking under partially visible kinematic chain: A unified approach," *IEEE Transactions on Robotics*, 2021.

[23] J. Lambrecht and L. Kästner, "Towards the usage of synthetic data for marker-less pose estimation of articulated robots in rgb images," in *2019 19th International Conference on Advanced Robotics (ICAR)*, pp. 240–247, IEEE, 2019.

[24] J. Lu, A. Jayakumari, F. Richter, Y. Li, and M. C. Yip, "Super deep: A surgical perception framework for robotic tissue manipulation using deep learning for feature extraction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4783–4789, IEEE, 2021.

[25] R. Hao, O. Özgüner, and M. C. Çavuşoğlu, "Vision-based surgical tool pose estimation for the da vinci® robotic surgical system," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1298–1305, IEEE, 2018.

[26] Y. Labbé, J. Carpentier, M. Aubry, and J. Sivic, "Single-view robot pose and joint angle estimation via render & compare," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1654–1663, 2021.

[27] 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.

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

[29] J. Denavit and R. S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," 1955.

[30] 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.

[31] J. A. Sethian, "Fast marching methods," *SIAM review*, vol. 41, no. 2, pp. 199–235, 1999.

[32] 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.

[33] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[34] E. Rohmer, S. P. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1321–1326, IEEE, 2013.

[35] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, IEEE, 2017.