2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)
April 4-8, 2022, Edinburgh, UK

Sensing soft robots' shape with cameras: an investigation on kinematics-aware SLAM

Emanuele Riccardo Rosi*,1,2, Maximilian Stölzle*,1, Fabio Solari[2], Cosimo Della Santina[1,3]

***Abstract*— The nature of continuum soft robots calls for novel perception solutions, which can provide information on the robot's shape while not substantially modifying their bodies' softness. One way to achieve this goal is to develop innovative and completely deformable sensors. However, these solutions tend to be less reliable than classic sensors for rigid robots. As an alternative, we consider here the use of monocular cameras. By admitting a small rigid component in our design, we can leverage well-established solutions from mobile robotics. We propose a shape sensing strategy that combines a SLAM algorithm with nonlinear optimization based on the robot's kinematic model. We prove the method's effectiveness in simulation and with experiments of a single-segment continuous soft robot with a camera mounted to the tip. We achieve mean relative translational errors below 9% simulations and experiments alike, and as low as 0.5% on average for some simulation conditions.**

I. Introduction

With their bodies entirely made of soft deformable materials, continuum soft robots are especially suited for application domains involving safe and robust interaction with humans and environment, and ranging from inspection, to healthcare and agriculture [1], [2]. To achieve these goals, soft robots must first master the art of controlling and sensing their body shape in space [3]. A major challenge with shape perception in soft robots is that sensing strategies must not compromise the intrinsic softness of these systems [4], [5]. To this end, researchers have proposed several entirely deformable sensors over the years, including capacitive [6], and optical sensors [7], liquid metal [8]. These solutions are quite attractive since they minimally corrupt the physical softness of the robot. However, they usually require complex learning strategies to be used since their behavior is hard to model [9], [10].

An alternative is to relax the constraint of complete deformability and allow from small rigid components. This strategy enables rethinking the use of existing sensing technologies in this radically new context. Examples are hall sensors [11], IMUs [12], and microphones [13]. The information gathered from inward-facing cameras looking at features in soft chambers' inner walls has proven sufficient to estimate the configuration [14], [15], and characterize contacts with the environment [16], [17]. However, all these strategies need machine learning to transform the image information in the desired physical quantity. Alternatively, cameras mounted outwards on the robot's tip and have been used to execute visual servoing [18]. In the best of Authors' knowledge, the only two works dealing with continuum (non-soft) robots are [19] and [20]. The first uses Bundle Adjustment (BA) to integrate the output of multiple cameras embedded in a single segment. The second uses hand-tuned features to estimate the robot configuration within a novel kinematic model. Thus, no general strategy to estimate the whole state of a soft segment from a single monocular camera exists in the literature.

Simultaneous Localization and Mapping (SLAM) is one of the most effective and largely used strategies for vision-based localization for mobile robots and autonomous vehicles [21], [22]. In this work, we investigate using monocular SLAM to estimate the location of selected points across the soft robot. We then propose a mechanism for simultaneously refining the estimation and reconstructing the complete shape of the robot. We do that by retracting the output of the SLAM to the manifold of camera configurations admitted by the kinematic model of the soft robot. We formulate this action as a nonlinear optimization problem. Fig. 1(a) summarizes the proposed architecture. We test the strategy with simulations and experiments, achieving mean relative translational errors between $0.4\,\%$ and $9\,\%$ in the former, and from $5\,\%$ to $9\,\%$ in the latter.

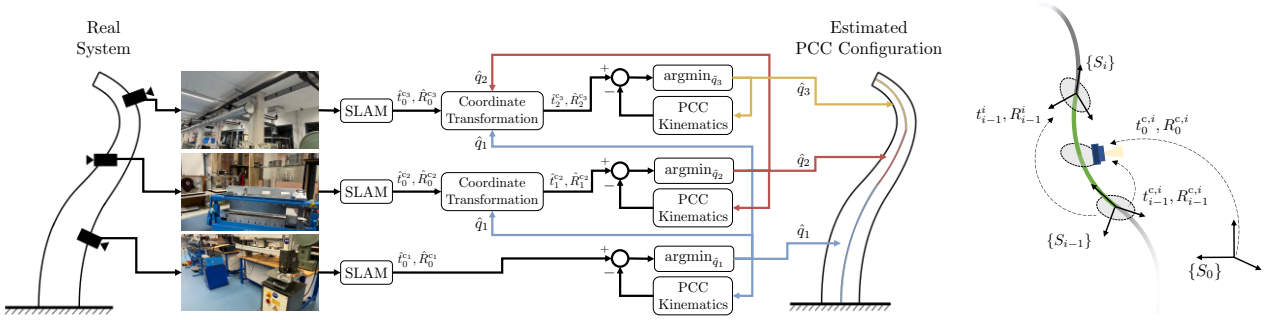
II. Proposed Architecture: Shape Estimation with Kinematics-Aware SLAM

In this work we propose to use SLAM for proprioception of continuum soft robots. Common kinematic parametrizations for soft robots such as Piecewise Constant Curvature (PCC) [23] or Piecewise Constant Strain (PCS) [24] model the soft arm to consist of multiple segments with independent kinematic state variables. In the following, we assume that the robot's model consists of n_{S} segments, and that a monocular camera is attached to each segment. We rely on a PCC kinematic formulation. However, the proposed results can be directly generalized to the PCS case. Our goal is to reconstruct the full shape (i.e., a configuration q_i for each segment) of the soft robot from the stream of images recorded by the cameras.

Fig. 1(a) shows an overview of the proposed architecture. We use SLAM algorithms such as monocular ORB-SLAM [22] to estimate the pose of cameras attached to the soft robotic arm. The pose estimation consists of 3D translation and rotations describing the relative camera movement from initial calibration to the current state ($\hat{t}_0^{\mathrm{c}_i}, \hat{R}_0^{\mathrm{c}_i}$

*Authors contributed equally. [1]Cognitive Robotics department, Delft University of Technology, Mekelweg 2, 2628 CD Delft, Netherlands. {M.W.Stolzle, C.DellaSantina}@tudelft.nl [2]Department of Computer Science, Bioengineering, Robotics and System Engineering (DIBRIS), University of Genoa, Viale Causa 13, 16145 Genoa, Italy s4361996@studenti.unige.it, fabio.solari@unige.it [3] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Weßling, Germany

2022 IEEE 5th International Conference on Soft Robotics (RoboSoft) | 978-1-6654-0828-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ROBOSOFT54090.2022.9762199



(a) Overview (b) Kinematics

Fig. 1: Panel (a) shows a pictorial representation of the proposed perception strategy. Cameras are attached to a soft continuum robot. We propose to use ORB-SLAM [22] to gather a pose estimate for each camera. The results are iteratively combined to extract local transformation, and refined by projecting the resulting postures onto the manifold of configurations attainable with the Piecewise Constant Curvature (PCC) kinematics. The result is an estimation of the full shape within the selected kinematic description $\hat{q}$. Panel (b) reports the main quantities of the kinematic model of one segment.

in the figure). Then, the kinematic model is simultaneously used to refine the outputs of the SLAM and transform it to the desired estimation of the configurations ($\hat{q}_1, \hat{q}_3, \hat{q}_3$ in the figure). Starting from the base segment, and progressively iterating up until reaching the tip of the robot, we express translation and rotations in local coordinates, and we optimize the estimated configurations segment-by-segment starting at the proximal end by projecting the pose estimate into the 3D PCC kinematics.

In the following subsections, we provide more details on the various components of this architecture.

A. Background: Monocular ORB-SLAM

We choose monocular ORB-SLAM [22] for pose estimation of the camera locations. Although this technological solution has never been applied to soft robots, the algorithm itself is well established and its applications to mobile robotics widespread. As such, we briefly describe here only the major steps of the algorithm.

1) **Map Initialization**: ORB-SLAM initializes a map of 3D points based on two video frames. The 3D points and relative camera pose are computed using triangulation of 2D ORB feature correspondences.
2) **Tracking**: Once the map is initialized, the camera pose is estimated for each new frame by matching features in the current frame to features in the last key frame. The estimated camera pose is refined by tracking the local map.
3) **Local Mapping**: If the current frame is identified as a key frame, it is used to create new 3D map points. At this stage, BA is used to minimize re-projection errors by adjusting the camera pose and 3D points.
4) **Loop Closure**: Loops are detected for each key frame by comparing it against all previous ones. This information is used to optimize the poses.

An important aspect of SLAM algorithms are key frames, which are a subset of video frames that contain cues for localization and tracking. Two consecutive key frames usually involve sufficient visual change.

B. Projection into PCC-Kinematics

Once the SLAM algorithm provides us with the estimated camera poses, we want to interpret and correct them such as that they are coherent with the PCC kinematic model. In this paper, we consider the *Delta* parametrization [25] of the PCC kinematics, but the formulation could also be easily adapted to other kinematic parametrizations such as PCS [24].

We show a pictorial representation of this kinematic model in Fig. 1(b). Each segment of original length $L_{0,i}$ is described with three configuration variables $q_i \in \mathbb{R}^3 = \begin{pmatrix}\Delta_{x,i} & \Delta_{y,i} & \delta L_i\end{pmatrix}^{\mathrm{T}}$, where δL_i is segment's extension, and $\Delta_{x,i}$ and $\Delta_{y,i}$ are the differences of the arc lengths of the segment at a radial distance of d_i from the center-line along both cardinal directions of the base [25]. The complete robot's configuration is $q \in \mathbb{R}^{3n_{\mathrm{S}}}$. The coordinate transformation for segment i from the base frame $\{S_{i-1}\}$ into the tip frame $\{S_i\}$ as a function of the configuration q_i is given by [25]

$$R^i_{i-1} = \begin{pmatrix} 1+\frac{\Delta^2_{x,i}}{\Delta^2_i}(\mathrm{c}_i-1) & \frac{\Delta_{x,i}\Delta_{y,i}}{\Delta^2_i}(\mathrm{c}_i-1) & \frac{\Delta_{x,i}}{\Delta_i}\mathrm{s}_i \\ \frac{\Delta_{x,i}\Delta_{y,i}}{\Delta^2_i}(\mathrm{c}_i-1) & 1+\frac{\Delta^2_{y,i}}{\Delta^2_i}(\mathrm{c}_i-1) & \frac{\Delta_{y,i}}{\Delta_i}\mathrm{s}_i \\ \frac{-\Delta_{x,i}}{\Delta_i}\mathrm{s}_i & \frac{-\Delta_{y,i}}{\Delta_i}\mathrm{s}_i & \mathrm{c}_i \end{pmatrix},$$
$$t^i_{i-1} = \frac{d_i(L_{0,i}+\delta L_i)}{\Delta^2_i}\begin{pmatrix}\Delta_{x,i}(1-\mathrm{c}_i) & \Delta_{y,i}(1-\mathrm{c}_i) & \Delta_i \mathrm{s}_i\end{pmatrix}^{\mathrm{T}}, \tag{1}$$

where we substituted $\Delta_i = \sqrt{\Delta^2_{x,i}+\Delta^2_{y,i}}$, $\mathrm{s}_i = \sin\left(\frac{\Delta_i}{d_i}\right)$, and $\mathrm{c}_i = \cos\left(\frac{\Delta_i}{d_i}\right)$ for conciseness.

We describe the coordinate frame of camera i with $\{S_{\mathrm{c}_i}\}$. It is assumed that there exists a fixed transformation $T^{\mathrm{c}_i}_{\check{\mathrm{c}},i} \in \mathbb{R}^{4\times4}$ from frame $\{S_{\check{\mathrm{c}},i}\}$ to the camera frame $\{S_{\mathrm{c}_i}\}$. $\{S_{\check{\mathrm{c}},i}\}$ is localized at a distance l_{c_i} along the center-line from the base of segment i. The transformation $T^{\check{\mathrm{c}},i}_{i-1}(q_{\check{\mathrm{c}},i})$ from the base to the frame $\{S_{\check{\mathrm{c}},i}\}$ can be found with (1) by plugging in the adjusted configuration $q_{\check{\mathrm{c}},i}$ defined as

$$q_{\check{\mathrm{c}},i} = \frac{l_{\mathrm{c}_i}}{L_{0,i}} q_i, \tag{2}$$

and the adjusted original length l_{c_i}.

The SLAM algorithm provides a pose estimate for the translation $\hat{t}^{\mathrm{c}_i}_{\mathrm{c,t0},i} \in \mathbb{R}^3$ and rotation $\hat{R}^{\mathrm{c}_i}_{\mathrm{c,t0},i} \in \mathbb{R}^{3\times3}$ relative to the known initial reference frame of the camera $\{S_{\mathrm{c,t0},i}\}$. Thus, we first transform the pose estimates to the inertial frame of the robot $\{S_0\}$

$$\begin{pmatrix} \hat{t}^{\mathrm{c}_i}_0 \\ 1 \end{pmatrix} = T_0^{\mathrm{c,t0},i} \begin{pmatrix} \hat{t}^{\mathrm{c}_i}_{\mathrm{c,t0},i} \\ 1 \end{pmatrix}, \quad \hat{R}^{\mathrm{c}_i}_0 = R_0^{\mathrm{c,t0},i} \hat{R}^{\mathrm{c}_i}_{\mathrm{c,t0},i}. \tag{3}$$

We introduce the following notations for the PCC kinematics

$$\begin{aligned} \hat{t}^{\mathrm{c}_i}_{i-1} &= \Pi_{\mathrm{t}}(\hat{q}_i) = \hat{T}^{\check{\mathrm{c}},i}_{i-1}\left(\frac{l_{\mathrm{c}_i}}{L_{0,i}}\hat{q}_i\right) t^{\mathrm{c}_i}_{\check{\mathrm{c}},i} \\ \hat{R}^{\mathrm{c}_i}_{i-1} &= \Pi_{\mathrm{R}}(\hat{q}_i) = \hat{R}^{\check{\mathrm{c}},i}_{i-1}\left(\frac{l_{\mathrm{c}_i}}{L_{0,i}}\hat{q}_i\right) R^{\mathrm{c}_i}_{\check{\mathrm{c}},i}, \end{aligned} \tag{4}$$

where $\hat{t}^{\check{\mathrm{c}},i}_{i-1}$ and $\hat{R}^{\check{\mathrm{c}},i}_{i-1}$ describe the translation and rotation from the base of the segment to the camera frame according to the PCC kinematic model for an estimated configuration of the the segment $\hat{q}_i$. $\hat{T}^{\check{\mathrm{c}},i}_{i-1}(q_{\check{c},i})$ and $\hat{R}^{\check{\mathrm{c}},i}_{i-1}(q_{\check{c},i})$ are based on (1) and a function of the adjusted configuration $q_{\check{c},i}$ referenced in (2).

Additionally, the pose estimates by the SLAM algorithm need to be transformed to the base frame of segment i. Thus, we introduce the following

$$\begin{aligned} \hat{t}^{\mathrm{c}_i}_{i-1} &= \Psi_{\mathrm{t}}(q_1 \dots q_{i-1}, \hat{t}^{\mathrm{c}_i}_0) = \prod_{\tilde{i}=1}^{i-1}\left(T^{\tilde{i}}_{\tilde{i}-1}(\hat{q}_{\tilde{i}})\right)^{\mathrm{T}} \begin{pmatrix} \hat{t}^{\mathrm{c}_i}_0 \\ 1 \end{pmatrix}, \\ \hat{R}^{\mathrm{c}_i}_{i-1} &= \Psi_{\mathrm{R}}(q_1 \dots q_{i-1}, \hat{R}^{\mathrm{c}_i}_0) = \prod_{\tilde{i}=1}^{i-1}\left(R^{\tilde{i}}_{\tilde{i}-1}(\hat{q}_{\tilde{i}})\right)^{\mathrm{T}} \hat{R}^{\mathrm{c}_i}_0. \end{aligned} \tag{5}$$

Next, we define a cost function to optimize the pose estimate by projecting them into the PCC-kinematics

$$\min_{\hat{q}} \sum_{i=1}^{n_{\mathrm{S}}} f_{\mathrm{t},i}(\hat{q}) + \lambda_{\mathrm{R}} f_{\mathrm{R},i}(\hat{q}), \tag{6}$$

with

$$\begin{aligned} f_{\mathrm{t},i}(\hat{q}) &= \left\|\Pi_{\mathrm{t}}(\hat{q}_i) - \Psi_{\mathrm{t}}(q_1 \dots q_{i-1}, \hat{t}^{\mathrm{c}_i}_0)\right\|_2, \\ f_{\mathrm{R},i}(\hat{q}) &= \left\|\Pi_{\mathrm{R}}(\hat{q}_i) - \Psi_{\mathrm{R}}(q_1 \dots q_{i-1}, \hat{R}^{\mathrm{c}_i}_0)\right\|_F, \end{aligned} \tag{7}$$

where the Euclidean norm is used to compute the translational error between the predicted translation by the PCC kinematic model and the estimated translation by SLAM. The rotational error is weighted with λ_{R} and computed with the Frobenius norm between the predicted rotation matrix by the PCC kinematics and the estimated orientation by SLAM represented as a rotation matrix as well.

Please note, that the optimization of the configuration estimate $\hat{q}$ can be decoupled for each segment. We start by optimizing the configuration of the first segment $\hat{q}_1$ based on $\Psi_{\mathrm{t}}(\hat{t}^{\mathrm{c},1}_0)$ and $\Psi_{\mathrm{R}}(\hat{R}^{\mathrm{c},1}_0)$. Next, we optimize the configuration of the second segment $\hat{q}_2$ taking into account the already optimized configuration of segment one in $\Psi_{\mathrm{t}}(\hat{q}_1, \hat{t}^{\mathrm{c},2}_0)$ and $\Psi_{\mathrm{R}}(\hat{q}_1, \hat{R}^{\mathrm{c},2}_0)$. Subsequently, we move on to optimize the remaining segments sequentially as described by Algorithm 1. This procedure is also graphically represented by the right side of Fig. 1(a).

Algorithm 1 Pose estimation for soft robots through SLAM

Input: $o \in \mathbb{R}^{n_{\mathrm{S}}}$ {Observations of all cameras}
Output: $\hat{q} \in \mathbb{R}^{3n_{\mathrm{S}}}$ {Estimated robot configuration}
  $i \leftarrow 1$
  while $i \leq n_{\mathrm{S}}$ **do**
    $\hat{t}^{\mathrm{c}_i}_0 \leftarrow T_0^{\mathrm{c,t0},i}\,\mathrm{SLAM_t}(o_i)$ {Translational est. SLAM}
    $\hat{R}^{\mathrm{c}_i}_0 \leftarrow R_0^{\mathrm{c,t0},i}\,\mathrm{SLAM_R}(o_i)$ {Rotational est. SLAM}
    $f_{\mathrm{t},i}(\hat{q}) \leftarrow \left\|\Pi_{\mathrm{t}}(\hat{q}_i) - \Psi_{\mathrm{t}}(q_1 \dots q_{i-1}, \hat{t}^{\mathrm{c}_i}_0)\right\|_2$
    $f_{\mathrm{R},i}(\hat{q}) \leftarrow \left\|\Pi_{\mathrm{R}}(\hat{q}_i) - \Psi_{\mathrm{R}}(q_1 \dots q_{i-1}, \hat{R}^{\mathrm{c}_i}_0)\right\|_F$
    $f_{c,i}(\hat{q}) \leftarrow f_{\mathrm{t},i}(\hat{q}) + \lambda_{\mathrm{R}} f_{\mathrm{R},i}(\hat{q})$ {Cost function for $\hat{q}_i$}
    $\hat{q}_i \leftarrow \mathrm{argmin}_{\hat{q}_i} f_c(\hat{q})$
    $i \leftarrow i + 1$
  end while

III. Simulations

We quantitatively evaluate our approach in simulation for one soft robotic segment with a camera attached to the tip of the robot. We first compute trajectories which behave according to PCC kinematics. Next, we render photo-realistic images for the camera attached to the tip of the segment for every time-step using a virtual environment implemented in Blender. Subsequently, we process the synthetic camera images with the ORB-SLAM [22] algorithm and projected the estimated poses of the tip of the segment into the PCC kinematic model as outlined in Section II. Finally, we compare the estimated poses against the ground-truth and statistically evaluate the Root-Mean-Square Error (RMSE) both for translational and rotational estimates. More details follow.

A. System

We consider a soft robotic segment of diameter $20\,\mathrm{mm}$ diameter and with varying lengths $L_{0,1}$ between $15\,\mathrm{cm}$ and $100\,\mathrm{cm}$. As the camera is attached to the tip of the segment, we set $l_{\mathrm{c}_1} = L_{0,1}$ and define $T^{\mathrm{c},1}_{\check{\mathrm{c}},1} = \mathrm{diag}(\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix})$.

B. Trajectories and calibration sequence

Three different trajectories are considered for the simulated movement of the soft robotic segment and its attached virtual camera. While the first one represents a planar side bending, the second one describes an "8" shape with the tip, and the last one covers a lobe of the "8". Those trajectories were commanded in $\Delta_{x,1}$ and $\Delta_{y,1}$, with the following mathematical formulas:

$$\Delta_{x,1} = L_{0,1} A_x \sin(2\pi f_x) \qquad \Delta_{y,1} = L_{0,1} A_y \sin(2\pi f_y), \tag{8}$$

with $L_{0,1}$ the unextended length of the robot, A_x and A_y amplitudes of the sinusoids and f_x and f_y frequencies of the sinusoids. The parameters f_x and f_y are defined as follows with k representing the current time index:

$$f_x = \frac{F_x(k-1)}{n_{\mathrm{t}}}, \qquad f_y = \frac{F_y(k-1)}{n_{\mathrm{t}}}. \tag{9}$$

Please note that these trajectories do not contain any elongation of the segment. We list the chosen amplitudes and frequencies of the trajectories in Table I. Each trajectory is generated considering different robot lengths, namely $15\,\mathrm{cm}$, $30\,\mathrm{cm}$ and $100\,\mathrm{cm}$. The number of time steps n_{t} is chosen at 120. The commanding of $\Delta_{x,1}$ and $\Delta_{y,1}$ is such that the amplitude and frequency of the trajectory are independent of the number of frames (i.e., time steps). In Figure 2 we

TABLE I: Parameters of the implemented trajectories. We list the amplitudes and the frequencies of the trajectories parametrized by $\Delta_{x,1}$ and $\Delta_{y,1}$ as specified in (8).

Trajectory	A_x	A_y	F_x	F_y
Trajectory 1: planar side bending	0.1	0	0.5	0
Trajectory 2: half 8-shape	0.05	0.05	1	0.5
Trajectory 3: full 8-shape	0.05	0.05	2	1

show a 3D visualization of the trajectories corresponding to a segment length of $15\,\mathrm{cm}$.

In addition to the robot trajectory, a calibration sequence trajectory is designed to initialize the SLAM map. It is good practice to move the camera parallel to the scene captured. Accordingly, we decide to move the camera into the x cardinal direction of segment base frame with the translation distance proportional to the robot length.

C. Rendering of synthetic images

The rendering software Blender allows us, among other things, to load a 3D model of the environment, follow customized trajectories with a virtual camera, and render photo-realistic synthetic pictures of the environment from the chosen camera perspective. We use an interior scene published by Nextwave Multimedia. We report a view of the scene in Fig. 2(d). The virtual camera is set to be perspective with a focal length of $30\,\mathrm{mm}$. For each run, we randomly initialise the trajectory at one of seven predefined launch points in the indoor environment to diversify the coverage of the environment. The x-, y-, and z-coordinates of the seven initial positions have a standard deviation of $0.5\,\mathrm{m}$, $0.2\,\mathrm{m}$, $0.4\,\mathrm{m}$ respectively. The initial orientation represented in XYZ Euler angles varies with a standard deviation of $0.05\,\mathrm{rad}$, $0.13\,\mathrm{rad}$, and $1.17\,\mathrm{rad}$. For each trajectory, we render 120 synthetic images along the trajectory and save them to a folder for later offline processing by the ORB-SLAM [22] algorithm. Fig. 3 reports a few representative stills of what the robot sees during one execution of the three trajectories discussed above.

D. Implementation of ORB-SLAM

The synthetic images along the trajectory are processed offline by the ORB-SLAM [22] algorithm. We rely on the official MATLAB implementation of ORB-SLAM. While we run our simulations offline to decouple any delays by the rendering and / or SLAM pipeline, we would like to point out that other ORB-SLAM implementations such as for example in C++ are able to be run in real-time at frame rates of between $10\,\mathrm{Hz}$ to $30\,\mathrm{Hz}$ [22].

E. Projection into PCC kinematics

The trade-off parameter λ_R between the rotational and the translational error in the cost function (6) was manually tuned and set to $\lambda_\mathrm{R} = 0.4$. As the simulations do not contain any elongations of the segment, we set $\delta L_1 = 0$ in (1). We solve the optimization problem outlined in (6) using nonlinear least-squares with the Levenberg-Marquardt solver.

F. Evaluation metrics

To quantitatively evaluate the performance of our proposed approach, we introduce error metrics for both the translation and orientation estimates. We measure the translational pose prediction error with a relative RMSE e_t

$$e_\mathrm{t} = \frac{\sqrt{\sum_{t=1}^{n_\mathrm{t}} \left(\|\hat{t}_{0,t}^{\mathrm{c},1} - t_{0,t}^{\mathrm{c},1}\|_2\right)^2}}{\sqrt{n_\mathrm{t}}\, l_\mathrm{traj}}, \tag{10}$$

where l_traj corresponds to the length of the trajectory and n_t the total number of data points along the trajectory. Similarly, we leverage the Frobenius norm for the rotational error e_R

$$e_\mathrm{R} = \sqrt{\sum_{t=1}^{n_\mathrm{t}} \frac{\left(\left\|\hat{R}_{0,t}^{\mathrm{c},1} - R_{0,t}^{\mathrm{c},1}\right\|_F\right)^2}{n_\mathrm{t}}}. \tag{11}$$

As torsion can often be neglected for soft robotic arms, we state the angle error for the orientation of the local z-axis of the tip of the segment for intuitive analysis of the orientation estimates. First, the unit vector of the local z-axis $\{o_1\}_0$ is computed in the base frame $\{S_0\}$

$$\{o_1\}_0 = R_{0,t}^1 \begin{pmatrix}0 & 0 & 1\end{pmatrix}^\mathrm{T}, \quad \{\hat{o}_1\}_0 = \hat{R}_{0,t}^1 \begin{pmatrix}0 & 0 & 1\end{pmatrix}^\mathrm{T}, \tag{12}$$

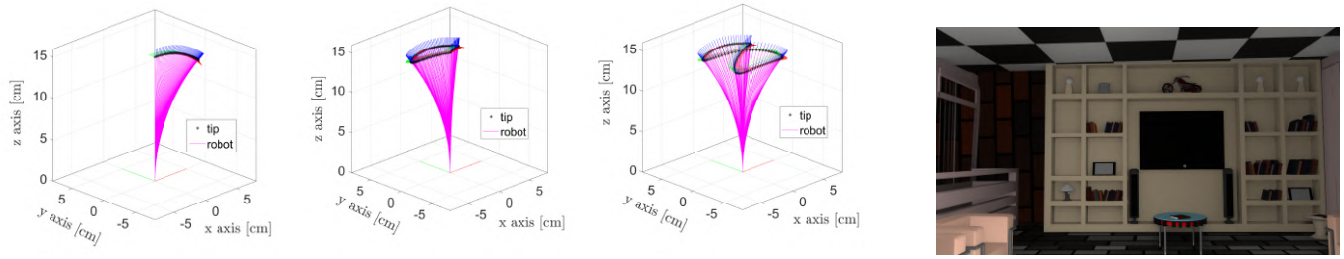
which allows us to subsequently compute the angle error between the ground-truth z-axis of the tip $\{o_1\}_0$ and the estimated z-axis $\{\hat{o}_1\}_0$

$$e_{\theta_z} = \sqrt{\sum_{t=1}^{n_\mathrm{t}} \frac{\left(\arccos\left(\{o_1\}_0 \cdot \{\hat{o}_1\}_0\right)\right)^2}{n_\mathrm{t}}}. \tag{13}$$

G. Results

We evaluate our proposed method in simulation on three different robot segment lengths ($15\,\mathrm{cm}$, $30\,\mathrm{cm}$, and $100\,\mathrm{cm}$) and for the three trajectories previously described. We state statistical results such as mean, standard deviation, lower and upper bounds over seven separate trials each covering a different part of the indoor environment. The errors are reported both for the SLAM estimates *before* optimization and *after* projection into the PCC kinematics. While the results for the relative RMSE of translation estimates through the entire trajectory are shown in Table II, the absolute RMSE of rotation matrices computed with the Frobenius norm of the rotation matrices or the z-axis orientation axis angle error are displayed in Table III.

Our results show translation errors of in average $6\,\%$ to $9\,\%$ for short segments and $2\,\%$ to $6\,\%$ RMSE relative to the trajectory length for long segments before optimization. The projection into PCC kinematics significantly decreases the translational error by between $66\,\%$ and $96\,\%$ to $0.4\,\%$ to $2\,\%$ for short segments and $0.6\,\%$ to $2\,\%$ for long segments. We state an absolute RMSE for the orientation estimates of the z-axis of the tip e_{θ_z} as defined in Eq. 13 of between $0.005\,\mathrm{rad}$ and $0.1\,\mathrm{rad}$ after optimization. The rotational error of the orientation estimates varies by trial but in average stays constant across the optimization. Choosing a bigger weight λ_R on the rotational loss during the optimization resulted in larger improvements for estimation the orientation at the cost of higher translational errors.



(a) Trajectory 1 (b) Trajectory 2 (c) Trajectory 3 (d) Virtual interior scene

Fig. 2: 3D visualization of the three trajectories used in our simulations and experiments for a segment with 15 cm length. In magenta we visualize the trajectory of the full robot, and in black the tip positions. Additionally, the tip orientation (red = x-axis, green = y-axis, blue = z-axis) is displayed. The virtual interior scene used for the Blender renderings is presented in the last column.

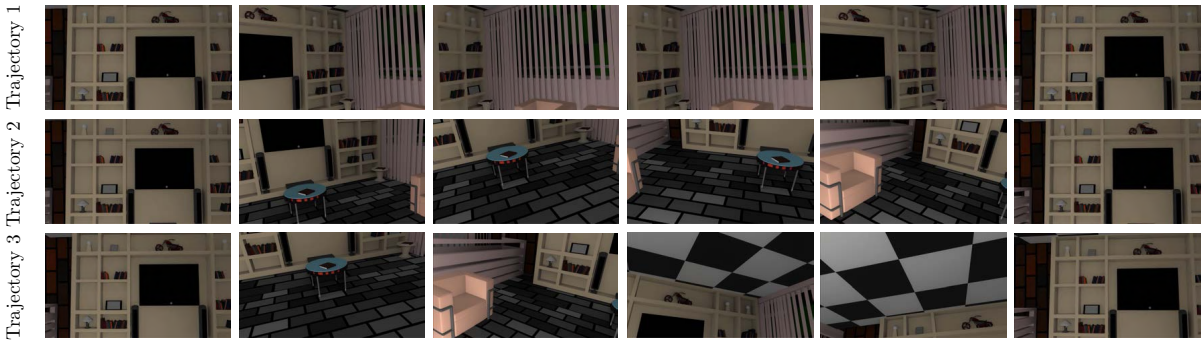


Fig. 3: Sequence of stills showing the rendered images by the virtual camera in Blender for three different trajectories and a robot segment of length 15 cm. The trajectories are visualized in Figure 2.

TABLE II: Relative RMSE [%] for translations as referenced in (10) of various trajectories and of robot segments with different lengths (15 cm, 30 cm, 100 cm). We state the error as mean $\pm$ stdev $(\min, \max)$ and compute the statistics over seven trials from different initial poses.

Trajectory	**Optimization**	$L_{0,1} = 15\,\mathrm{cm}$	$L_{0,1} = 30\,\mathrm{cm}$	$L_{0,1} = 100\,\mathrm{cm}$
Trajectory 1	No	$9 \pm 3\ (5, 12)$	$7 \pm 3\ (3, 13)$	$3 \pm 2\ (1, 7)$
Trajectory 1	Yes	$0.4 \pm 0.2\ (0.3, 0.8)$	$2 \pm 1\ (0, 4)$	$1.0 \pm 0.7\ (0.5, 2.3)$
Trajectory 2	No	$9 \pm 4\ (4, 17)$	$6 \pm 2\ (3, 8)$	$1.9 \pm 0.9\ (1.0, 3.0)$
Trajectory 2	Yes	$0.7 \pm 0.7\ (0.3, 1.8)$	$0.7 \pm 0.4\ (0.2, 1.2)$	$0.6 \pm 0.3\ (0.3, 0.9)$
Trajectory 3	No	$6 \pm 5\ (3, 16)$	$2.6 \pm 0.6\ (1.7, 3.3)$	$6 \pm 14\ (1, 37)$
Trajectory 3	Yes	$2 \pm 3\ (0, 9)$	$0.5 \pm 0.3\ (0.1, 0.8)$	$2 \pm 5\ (0, 15)$

TABLE III: Rotational errors of various trajectories and for robot segments with different lengths (15 cm, 30 cm, 100 cm). We report both an absolute RMSE computed with the Frobenius norm between the rotation matrices as stated in (11) and an angle error [rad] for the orientation of the z-axis of the tip of the segment as defined in (13). We state the error as mean $\pm$ stdev and compute the statistics over seven trials from different initial poses.

Trajectory	**Optim.**	$L_{0,1} = 15\,\mathrm{cm}$		$L_{0,1} = 30\,\mathrm{cm}$		$L_{0,1} = 100\,\mathrm{cm}$	
		e_{R}	e_{θ_z} [rad]	e_{R}	e_{θ_z} [rad]	e_{R}	e_{θ_z} [rad]
Trajectory 1	No	0.010 ± 0.004	0.005 ± 0.002	0.016 ± 0.007	0.009 ± 0.005	0.027 ± 0.017	0.02 ± 0.01
Trajectory 1	Yes	0.007 ± 0.002	0.005 ± 0.002	0.012 ± 0.007	0.008 ± 0.005	0.027 ± 0.019	0.02 ± 0.01
Trajectory 2	No	0.02 ± 0.02	0.01 ± 0.02	0.011 ± 0.006	0.006 ± 0.002	0.015 ± 0.008	0.009 ± 0.005
Trajectory 2	Yes	0.02 ± 0.02	0.01 ± 0.02	0.016 ± 0.009	0.005 ± 0.002	0.019 ± 0.008	0.013 ± 0.006
Trajectory 3	No	0.1 ± 0.2	0.1 ± 0.1	0.010 ± 0.002	0.006 ± 0.001	0.2 ± 0.4	0.1 ± 0.2
Trajectory 3	Yes	0.1 ± 0.2	0.1 ± 0.1	0.02 ± 0.01	0.005 ± 0.002	0.2 ± 0.3	0.1 ± 0.2

IV. Experiments

We confirm the simulation results in a preliminary experimental study by mounting a Raspberry Pi camera to the tip of a soft segment [26]. The robotic segment is guided to follow three trajectories similar to the ones tested in simulation (see Section III-B). A Motion-capture (MoCap)

setup is employed to gather an accurate ground-truth on the shape of the segment.

A. Experimental setup

We show the experimental setup in Figure 4(a). We consider a soft robotic silicone segment consisting of four independently inflatable cavities. The segment has a cylindrical shape with a length $L_{0,1}$ of $11\,\mathrm{cm}$ and a radius d_1 of $21\,\mathrm{mm}$. A 3D-printed ring with four motion capture markers is located near the tip of the segment. We mount a Raspberry Pi camera module v2.0 to the tip of the segment. This camera has a $8\,\mathrm{MP}$ sensor and records frames at a sampling rate of $30\,\mathrm{Hz}$ and a resolution of 1080p. The focal length is $3.04\,\mathrm{mm}$ and the field of view is $62.2° \times 48.8°$. The camera module is attached to a Raspberry Pi 3B+ single-board computer which saves the frames for later processing by the ORB-SLAM [22] algorithm. The camera is screwed onto a custom 3D-printed holder which in turn is glued with the tip plane of the segment. The segment with its four air chambers is actuated with a proportional pressure regulator. Tubing attached to the base of the segment connects each chamber with the assigned pneumatic valve of the pressure regulator. The segment is attached in up-side-down configuration to the top plane of a cubical cage of $750\,\mathrm{mm}$ side length. For a straight segment, the camera is facing downwards towards the floor of the cage, which is covered by multiple printed checkerboard patterns. We acquire ground-truth pose information of the tip of the segment using an Optitrack MoCap system. The ground-truth poses of the tip of the segment are recorded at $30\,\mathrm{Hz}$. We also include the elongation of the segment δL_1 in the cost function (6) of our optimization. To resemble the calibration sequence from simulation for the SLAM map, we manually move the robot lateral into the x-coordinate direction before fixing it to the cage for the start of the experiments.

B. Results

Our experimental results reported in Table IV and visualized for trajectory 3 in Figure 5 show translational relative RMSE of between $9\,\%$ and $20\,\%$ for the three trajectories before optimization. The orientation of the z-axis of the tip is estimated with a mean error of approximately $0.075\,\mathrm{rad}$. The translational error is improved to between $5\,\%$ and $9\,\%$ after projection into the PCC-kinematics. The optimization also slightly improves the rotational RMSE by $4\,\%$ to $10\,\%$ relative to naive SLAM.

The experimental results of the SLAM algorithm are coherent with the simulations as the small segment length ($11\,\mathrm{cm}$) used in the experiments increases the translational errors as shown similarly in the simulations for a robot of length $15\,\mathrm{cm}$. Even though the translational error is greatly reduced through optimization, it is still significantly higher than in simulation. Two reasons for this difference could be that a) the segment in simulation was modelled as in-extensible, while the real robot segment is extended via pneumatic pressurization, which introduces additional errors by SLAM not accurately estimating the elongation movement and b) the real robot does not perfectly behave according to the Constant Curvature (CC) approximation as the simulated robot does.

TABLE IV: Real-world results before and after optimization. The translational errors are stated through a relative RMSE as described in (10). For rotation, we report both an absolute RMSE computed with the Frobenius norm between the rotation matrices as stated in (11) and an angle error [rad] for the orientation of the z-axis of the tip of the segment as defined in (13). The results are averaged over two trials for each trajectory.

Error category	**Opt.**	**Traj. 1**	**Traj. 2**	**Traj. 3**
Translation e_t	No	$20.3\,\%$	$14.2\,\%$	$9.1\,\%$
Translation e_t	Yes	$9.1\,\%$	$8.9\,\%$	$5.0\,\%$
Rotation e_R	No	0.145	0.103	0.126
Rotation e_R	Yes	0.130	0.099	0.120
Rotation e_{θ_z}	No	$0.079\,\mathrm{rad}$	$0.068\,\mathrm{rad}$	$0.084\,\mathrm{rad}$
Rotation e_{θ_z}	Yes	$0.080\,\mathrm{rad}$	$0.067\,\mathrm{rad}$	$0.084\,\mathrm{rad}$

V. Conclusion

This paper investigated using a monocular camera in shape sensing of continuum soft robots, with the ultimate goal of implementing precise and reliable estimations at the cost of introducing small rigid parts into the hardware design. The contribution of this paper has been twofold. First, we proposed to use monocular SLAM with a soft robot. Second, we propose a regularization of the estimation based on a nonlinear projection in the manifold of admitted configuration. A nonlinear optimization implements the latter based on the kinematic model of the robot. We have performed extensive simulations with rendered images in Blender and lab experiments with a single segment soft robot. The nonlinear optimization based on the robot's kinematic model led to a significant improvement in translations and a marginal improvement in rotations. Future work will focus on extending the experimental validation of the method to multiple segments and cameras, bettering the SLAM by feeding back the kinematic projection in its state and using this estimation to implement closed-loop control. While we conducted our experiments in a lab environment under ideal conditions with the camera pointed at checkerboard patterns thus resulting in plenty of image features for SLAM to track, future work should investigate whether deployment environments for soft robots would be sufficiently feature-rich for the use of our proposed method.

References

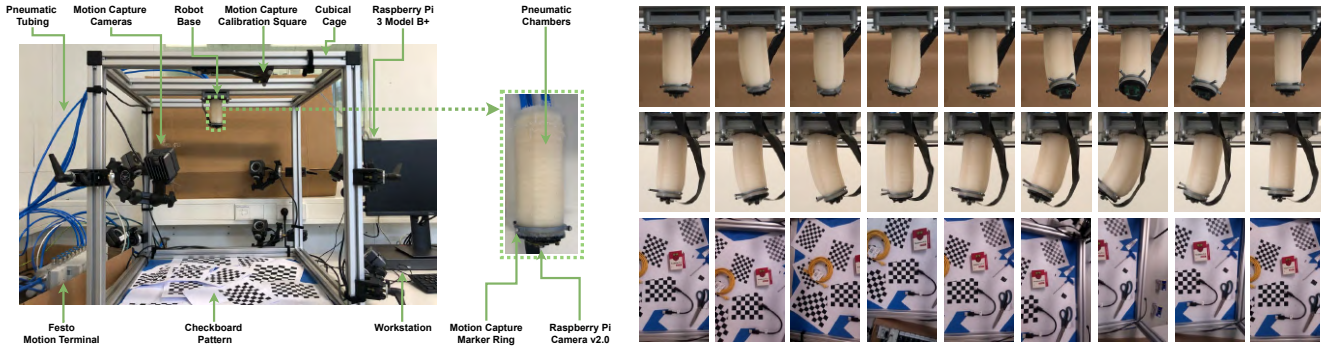
[1] C. Majidi, "Soft robotics: a perspective—current trends and prospects for the future," *Soft robotics*, vol. 1, no. 1, pp. 5–11, 2014.

[2] J. F. Elfferich, D. Dodou, and C. Della Santina, "Soft robotic grippers for crop handling or harvesting: A review," *Advanced Intelligent Systems*, 2021 (Under Review). [Online]. Available: https://edu.nl/frend

[3] C. Della Santina, C. Duriez, and D. Rus, "Model based control of soft robots: A survey of the state of the art and open challenges," *arXiv preprint arXiv:2110.01358*, 2021.

[4] P. Polygerinos, N. Correll *et al.*, "Soft robotics: Review of fluid-driven intrinsically soft devices; manufacturing, sensing, control, and applications in human-robot interaction," *Advanced Engineering Materials*, vol. 19, no. 12, p. 1700016, 2017.

[5] H. Wang, M. Totaro, and L. Beccai, "Toward perceptive soft robots: Progress and challenges," *Advanced Science*, vol. 5, no. 9, p. 1800541, 2018.



(a) Experimental setup

(b) Trajectory 3

Fig. 4: In Panel (a), a soft robotic segment is mounted to a cage with attached motion capture cameras. The segment is pneumatically actuated by a pressure regulator (Fest Motion Terminal). A Raspberry Pi camera v2.0 is attached to the tip of the segment and in a straight segment configuration looks down towards check-board patterns. Panel (b) depicts a sequence of stills showing the robot following trajectory 3 from two different vantage points. The second vantage point differs 90° from the first one. The third row displays a few representative frames as recorded by the camera attached to the tip of the segment.

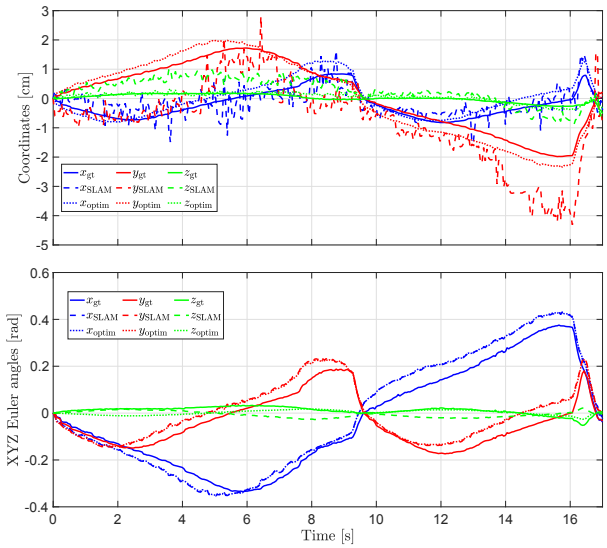


Fig. 5: Experimental results for trajectory 3. Comparison between ground-truth (solid line), SLAM (dashed line) and optimized through projection into PCC kinematics (dotted line) for translation and orientation estimates.

[6] L. Scimeca, J. Hughes *et al.*, "Model-free soft-structure reconstruction for proprioception using tactile arrays," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2479–2484, 2019.

[7] S. Li, S. A. Awale *et al.*, "Scaling up soft robotics: A meter-scale, modular, and reconfigurable soft robotic system," *Soft Robotics*, 2021.

[8] V. Wall, G. Zöller, and O. Brock, "A method for sensorizing soft actuators and its application to the rbo hand 2," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4965–4970.

[9] T. G. Thuruthel, B. Shih *et al.*, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Science Robotics*, vol. 4, no. 26, 2019.

[10] R. L. Truby, C. Della Santina, and D. Rus, "Distributed proprioception of 3d configuration in soft, sensorized robots via deep learning," *IEEE Robotics anamur2017orbd Automation Letters*, vol. 5, no. 2, pp. 3299–3306, 2020.

[11] H. Guo, F. Ju *et al.*, "Continuum robot shape estimation using permanent magnets and magnetic sensors," *Sensors and Actuators A: Physical*, vol. 285, pp. 519–530, 2019.

[12] J. Hughes, F. Stella *et al.*, "Sensing soft robot shape using imus: An experimental investigation," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 543–552.

[13] G. Zöller, V. Wall, and O. Brock, "Acoustic sensing for soft pneumatic actuators," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6986–6991.

[14] Y. She, S. Q. Liu *et al.*, "Exoskeleton-covered soft finger with vision-based proprioception and tactile sensing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 075–10 081.

[15] P. Werner, M. Hofer *et al.*, "Vision-based proprioceptive sensing: Tip position estimation for a soft inflatable bellow actuator," 2020.

[16] B. Ward-Cherrier, N. Pestell *et al.*, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft robotics*, vol. 5, no. 2, pp. 216–227, 2018.

[17] X. Lin, L. Willemet *et al.*, "Curvature sensing with a spherical tactile sensor using the color-interference of a marker array," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 603–609.

[18] B. S. Homberg, R. K. Katzschmann *et al.*, "Robust proprioceptive grasping with a soft robot hand," *Autonomous Robots*, vol. 43, no. 3, pp. 681–696, 2019.

[19] B. Weber, P. Zeller, and K. Kühnlenz, "Multi-camera based real-time configuration estimation of continuum robots," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3350–3355.

[20] H. Cheng, H. Liu *et al.*, "Approximate piecewise constant curvature equivalent model and their application to continuum robot configuration estimation," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 1929–1936.

[21] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial intelligence review*, vol. 43, no. 1, pp. 55–81, 2015.

[22] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[23] R. J. Webster III 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.

[24] F. Renda, F. Boyer *et al.*, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, 2018.

[25] C. Della Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1001–1008, 2020.

[26] R. K. Katzschmann, C. Della Santina *et al.*, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 454–461.