ViSE: Vision-Based Online Shape Estimation of Deformable Robots

Huhui Zheng*1,2, Sebastian Pinzello*1, Barnabas Gavin Cangan1, Thomas Buchner1, Robert K. Katzschmann1

Abstract—The precise control of soft and continuum robots requires knowledge of their shape, which has, in contrast to classical rigid robots, infinite degrees of freedom. To partially reconstruct the shape, proprioceptive techniques use built-in sensors resulting in inaccurate results and increased fabrication complexity. Exteroceptive methods so far rely on expensive tracking systems with reflective markers placed on all components, which are infeasible for deformable robots interacting with the environment due to marker occlusion and damage. Here, we present a regression approach for 3D shape estimation using a convolutional neural network. The proposed approach takes advantage of data-driven supervised learning and is capable of online marker-less shape estimation during inference. Two images of a robotic system are taken simultaneously at 25 Hz from two different perspectives, and are fed to the network, which returns for each pair the parameterized shape. The proposed approach outperforms marker-less state-of-the-art methods by a maximum of 4.5% in estimation accuracy while at the same time being more robust and requiring no prior knowledge of the shape. The approach can be easily implemented due to only requiring two color cameras without depth and not needing an explicit calibration of the extrinsic parameters. Online evaluations on two types of soft robotic arms and a soft robotic fish demonstrate our method’s accuracy and versatility on highly deformable systems. **

I. INTRODUCTION

Soft robots are experiencing a steep rise in popularity thanks to their ability to solve challenges such as compliant grasping and dexterous movement [1], [2], tasks with which rigid robots typically struggle [3]. To fully exploit the capabilities of soft robots, modern control approaches are needed, which typically rely on rich state feedback. However, obtaining and accurately describing the state of a continuously deforming soft body or robot is challenging compared to the state of a rigid object or robot. Encoders at the connecting joints of rigid robots readily provide precise state measurements, while soft robots mostly consist of elastomeric materials that deform with infinite degrees of freedom [4]. It is, therefore, crucial to solve the challenge of soft robotic state estimation to exploit the full potential of the great variety of soft robots for manipulation and beyond.

To date, various shape estimation approaches have attempted to improve soft robotic sensing capabilities. One type of sensing approach uses mechanical proprioception similar to classical robotic state estimation [5], [6]. A variety of sensor types such as resistive, capacitive, optical, and pneumatic transducers proprioceptively estimate the continuous deformations of soft robots [7], [8]. Mechanical proprioception with built-in sensors is limited by spatial resolution

*Equal contribution. Work done at Soft Robotics Lab, ETH Zürich.
**All code available on https://github.com/srl-ethz/vise.

Correspondence hehui.zheng@ai.ethz.ch
1Soft Robotics Lab, ETH Zürich, Zürich, Switzerland;
2ETH AI Center; Zürich, Switzerland.
capture in the tasks of estimating the shape of soft robotic arms and soft robotic fish.

Specifically, our work provides the following contributions:

- We present a simple-to-implement approach using two cameras that is applicable to various soft robots.
- We conduct a performance comparison of different CNN architectures and present the evaluation results on three real-world soft robots.
- We demonstrate the online estimation capability of our system, which enables its use for downstream applications such as the closed-loop control of soft robots.

Section II summarizes related work and Section III presents our methodology. In Section IV, we provide evaluation results to exhibit the performance of the proposed CNN-based approach and discuss the estimation accuracy under different model assumptions. Finally, Section V concludes the work and outlines directions for future research.

II. RELATED WORKS

Previous works demonstrate vision-based continuous shape estimation approaches that either work only for specific setups or necessitate strict image requirements, e.g., exact segmentation for contour extraction. In the following, we briefly discuss the most related works, focusing on marker-less shape estimation approaches.

Hannan and Walker use basic image processing techniques, including thresholding and image segmentation, to estimate the 2D shape of a planar, cable-actuated elephant trunk manipulator from single images [14]. However, their estimation results are only compared to another cable-based shape estimation technique but not with actual ground truth. Camarillo et al. extend the computer-vision methods to 3D shape estimation of a thin continuum manipulator [15]. If the precise positions of cameras are known, they could extract silhouettes from multiple cameras’ views, project those silhouettes into a volumetric space to find their intersection, and fit a spline through the resulting 3D point cloud. This approach requires a strong contrast between the tracked shape and the background, as well as the absence of other objects in the field of view.

Strict requirements on the image data are also found in other works. AlBeladi et al. rely on successful contour extraction of their soft arm to fit a geometric strain-based model to these edges [10]. Croom et al. also perform edge detection, but then fit reference points to the edges by using an unsupervised learning algorithm called the self-organizing map [16]. All of these approaches show good estimation results but require a strong contrast between the tracked object and the background.

Vandini et al. extract and join straight lines from a monoplane fluorescent surgical image to estimate the shape of a concentric tube robot [18]. By posing conditions for connecting the line features, they manage to relax image requirements and can extract curves from more unclean image data compared to the aforementioned works [19]. Reiter et al. [20] take on a similar approach to ours in that they extract features from segmented binary stereo-images. Since their feature extraction relies on the color-coded segments of their continuum robot, it does not generalize to other robots that do not have those features.

Mathis et al. created a deep learning framework based on transfer learning for marker-less pose estimation and tracking called DeepLabCut [21]. Their framework enables tracking of multiple visual features in unprocessed videos using only a small number of labeled frames for training. They demonstrate their method by tracking body parts of mice and show that they achieve pixel tracking errors comparable to human-level labeling accuracy. However, this framework by itself is restricted to pixel tracking in an image, and it cannot directly track 3D coordinates of features.

Our approach for 3D shape estimation of continuously deformable robots employs convolutional neural networks. While there are many vision-based proprioceptive methods for soft robots using deep learning [22]–[24], we focus on exteroceptive approaches that are simple to implement and do not add complexity to the manufacturing of the soft robots.

III. METHODS

Our proposed shape estimation method is a learned multi-view shape estimation from grayscale images. First, different views of the desired object are captured by two cameras. Then, unnecessary information in the images is removed by an image processing pipeline that transforms them into binary images. The processed images are then fed into a convolutional neural network (CNN) trained to estimate the parameters of the shape representation. The approach is outlined in Figure 1 and the following subsections detail the sub-components of our method.

A. Image Preprocessing

The RGB images of the shape are preprocessed to facilitate the learning procedure. The original images are converted to grayscale, cropped around the shape, and scaled to the size of 256 × 256 pixels. This preprocessing preserves enough information to accurately represent the shape, while keeping the number of parameters of the CNN relatively small. A median filter with a 7 × 7 pixel kernel size is applied to reduce noise before using adaptive thresholding to reduce the grayscale image to a binary image [25]. This step removes the background variations while preserving the shape. In addition, the adaptive nature of the thresholding operation and the resulting binary images make our trained network function in a wide range of lighting conditions without the need for retraining. Erosion and dilation with a 7 × 7 pixel kernel are applied for three iterations each to remove the remaining artifacts.

B. Shape Representations

We consider two different shape representations, the point, and piecewise constant curvature (PCC) model. Ground truth shape is obtained using the motion capture software (Qualysis Track Manager). Virtual coordinate frames are placed at the center of each group of motion markers to line up with the corresponding segment’s centroid. These coordinate frames allow the tracking of not only each segment’s translation but also orientation. The point representation employs
only the translation and comprises the positions of the key points along the soft robots, requiring three parameters for each key point. The PCC model [26], [27], a commonly used kinematic reduction model in soft robotics, can be fitted to both the translation and orientation of the virtual coordinate frames, allowing the modeling of a continuous shape by approximating it with multiple constant-curvature sections of fixed length. Each section is defined by two parameters, the curvature and an angle indicating the curving direction. Hence, compared to the point representation, the PCC model requires fewer parameters, 6 instead of 9 for a three-section soft arm, while representing a long, continuously deformable shape, which is useful for model-based control purposes.

C. Network Architecture

In this work, we compare the performance of the following CNN architectures: VGG [28] (VGG11, VGG11_bn with batch normalization, VGG13, VGG13_bn with batch normalization), ResNet [29] (ResNet18, ResNet34, ResNet50), EfficientNet [30] (EffNet_B0, EffNet1_B1, EffNet1_B3, EffNet1_B5), and EfficientNetV2 [31] (EffNetV2_s, EffNetV2_m). For all architectures, we modify the input channels to 2 and reduce the output size to 6, 9, 12, or 18, depending on the shape representation and number of sections to estimate. The output sizes can be found in Table I.

To avoid overfitting with large networks, we also introduce a custom truncated network, VGG_s_bn, adapted from the VGG architecture. Several convolutional layers are removed from the standard VGG network to further reduce the computational demand and improve online performance. The final soft-max layer is also removed to perform a regression instead of a classification. The network architecture is illustrated in Figure 2.

The network’s main elements are convolutional layers, batch normalization, rectification (ReLU) nonlinearities, and max pooling operations. These elements are applied in the mentioned order and repeated five times before the output is fed into two fully connected layers. All five convolutional layers have a kernel of size three, a stride of one, and a padding of one. The number of channel dimensions is increased from 2 to 6, 16, 32, 64, and 128. Batch normalization is applied before each convolutional layer. Every max-pooling operation reduces its input by a factor of two, reducing the initial image size of 256 × 256 pixels to 8 × 8 pixels after five operations. Hence, the input to the first fully connected layer is of size 8 × 128. A ReLU nonlinearity is applied after the first fully connected layer. The input to the last fully connected layer has a size of 1000, which is reduced to the output size of 6, 9, 12, or 18 (Table I).

D. Camera Realignment

No explicit camera calibration is needed and the camera configuration is implicitly learned by the CNN. This design choice limits the cameras’ positions to be fixed relative to the soft robot during the data collection. To alleviate the need for retraining, fiducial markers (AprilTags) [32] are attached to the robot’s base. The camera’s translation and rotation relative to the base can be extracted from the image of the fiducial markers. Our realignment utility for the camera pose compares the camera’s current and previously saved positions relative to the fiducial markers. With this utility, users can set up the RGB cameras close to the configuration during data collection and reuse the trained CNN repeatedly.

While this supervised approach still requires a motion capture setup to initially collect the ground truth data for training, the cameras can be moved without requiring retraining. A user can realign their cameras to approximately match the original poses relative to the robot’s base and still perform inference using the original training data. Given our realignment utility, the trained model still successfully estimates the shape of the robot without requiring a motion capture system.

IV. REAL-WORLD EXPERIMENTS

A. Experimental Setup

1) Soft Robots for Evaluation: The approach is tested on two types of soft robotic arms [12], [33] and a soft robotic fish [34] (Fig. 3). The first soft robotic arm (which we shall refer to as the WaxCast arm) consists of three axially connected cylindrical segments, each with four separately inflatable chambers. They are inflated using air provided through a pressure-controlled valve array (Festo SE & Co.
KG). By inflating one side, the chambers on that side elongate and induce bending in the segment, thus, the bending direction of the arm can be chosen by selecting the corresponding combination of inflation chambers. Each segment has a length of about $110 \pm 1$ mm and a diameter of $40 \pm 1$ mm. The combined length of the arm is $335 \pm 3$ mm. The second arm, SoPrA, is a two-segment soft robotic arm with fiber-reinforced pneumatic actuators. Segments are made of three individually fiber-reinforced elastomer air chambers that are glued together. Combining two of these segments adds up to a total length of $268 \pm 2$ mm. The robotic fish tail is similar in construction and actuation compared to the WaxCast arm, except that it is shaped like a fishtail. It consists of two inflatable chambers and has a total length of $115 \pm 1$ mm.

2) Data Collection: The ground truth data for learning is obtained by eight Mipus M3 motion-capture cameras from Qualisys AB placed in the motion capture space of $1.6 \times 1.1 \times 0.8$ m. The placement of the motion-capture markers can be seen in Figure 3. A group of reflective markers is placed on the arm and the other with six, which also contain visual markers placed in the motion capture space of $8 \times 8 \times 1$ m. Marker position data is supplied between them. Marker position data is supplied at $100$ Hz with an average accuracy of $0.1 \pm 0.2$ mm, while RGB image data is recorded at $25$ Hz by two depth cameras (Intel RealSense D435i) [35]. During the data acquisition, the pose data (i.e., the specific shape configurations of the soft robots) is collected such as to cover the robots’ full workspace. The segments of the WaxCast arm are all actuated to perform a circular motion, with periods of $100$, $10$, and $1$ second, for segments $1$, $2$, and $3$, respectively. We created two data sets, one with three motion-capture marker rings on the arm and the other with six, which also contain visual features in the form of black stripes that were put on the arm (Figure 3B). This process was repeated for SoPrA, but with the chambers randomly actuated. In total, three labeled data sets are generated for each arm, containing $12’000$ poses. The soft robotic fish is actuated to perform a tail-fin stroke with maximal deflection, resulting in a data set of $1’800$ poses. Each data set is split into $90\%$ training and $10\%$ testing sets.

3) Network training: The network is implemented and trained using the PyTorch framework. AdamW [36] is chosen as an optimizer and used to minimize the mean absolute loss. The network is trained per robot using a batch size of $64$ with an early stop for a maximum of $450$ epochs on each data set. The learning rate is set to $10^{-4}$ and reduced by $0.5$ after each $200$th epoch. Dropout is applied with a probability of $0.5$ in the fully connected layers during training to avoid overfitting. Training on a GPU (Nvidia GeForce RTX 3090) requires between $30$ to $60$ minutes to converge.

B. Results

| Exp. | Rep. - Sec. | Out. | Feat. | Point 1 | Point 2 | Point 3 |
|------|-------------|------|-------|---------|---------|---------|
| a    | PCC-6       | 12   | Yes   | 0.99% ± 0.3% | 3.2% ± 1.4% | 6.1% ± 2.4% |
| b    | PCC-3       | 6    | No    | 1.3% ± 0.6% | 3.6% ± 1.3% | 6.8% ± 3.5% |
| c    | Point-3     | 18   | Yes   | 0.06% ± 0.03% | 0.1% ± 0.07% | 0.4% ± 0.3% |
| d    | Point-3     | 9    | No    | 0.4% ± 0.4% | 0.8% ± 1.1% | 3.6% ± 5.0% |
| e    | Point-3     | 9    | Yes   | 0.06% ± 0.03% | 0.1% ± 0.06% | 0.3% ± 0.2% |

1) Shape Representations: The CNN was trained using the image data and either learned to output parameters of a PCC model that was fitted to the ground truth marker data or virtual marker positions along the arm (point estimation). We also analyze the approach’s accuracy when estimating just three PCC sections or virtual points compared to estimating six PCC sections or points. Both PCC and point estimation approaches were tested using the data sets from our two WaxCast arms (Figure 4a-d). Detailed results of the evaluation using VGG_s_bn can be seen in Table I for Experiments a-d, with the point estimation approach strictly outperforming the PCC approach. The errors are normalized based on the robot’s length, which is $335 \text{ mm} \pm 3 \text{ mm}$ for the soft arm.
2) **Visual Features:** To evaluate the effect of visual features on the estimation accuracy, the point estimation approach is applied to a data set with features (Figure 4e). We modified the WaxCast arm’s appearance to have multiple black stripes perpendicular to the arm’s backbone (Figure 3b). In Table I, Experiments d and e show that the mean tip error for the feature-less WaxCast arm is 3.6 ± 5.0% and only 0.3 ± 0.2% for the arm with features.

3) **CNN Architectures:** We tested 14 different CNN architectures and reported their performance on three soft robots in Table II. All errors are normalized with the corresponding robot’s length. The main performance metrics are mean and maximum tip estimation errors of the testing data set. Considering the limited data size, we prefer networks with better generalizability outside of the training data set. Therefore, we also include the mean tip estimation error of the training data set and the ratio of the training over test mean errors. To avoid possible overfitting to the training set, for each data set, the best CNN performance is picked among architectures with the overfit ratio under 1.5. Above this threshold, the testing error is more than 1.5 times the training error, suggesting an unbalanced performance not generalizable throughout the whole workspace. The best estimation performance for the WaxCast arm is 0.3% (1.01 mm) by VGG_s_bn, for SoPra, it is 0.54% (1.46 mm) by VGG13, and for the soft robotic fish, it is 0.62% (0.72 mm) by EffNetV2_m. We also report the parameter numbers with output size 6 and the average CNN forward frequency for a single estimation on a cluster GPU (NVIDIA Tesla V100-SXM2 32 GB).

4) **Benchmarks:** We compared the results of our point estimation approach with four similar works (see Table III), which also estimated reference points along continuously deforming shapes. Since the previous works either are shape agnostic [10], [15], [18] or employ active LED markers [37], they are unsuitable for our application with different soft robots. Due to limited data and the lack of open-source code, we only compared the tip errors, which are usually the largest and normalized with the length of each corresponding robot for a fair comparison. We believe that achieving low tip position reconstruction error is important for soft robotic shape estimation methods since this accuracy is critical in real-world operations involving reaching and grasping of objects.

**DeepLabCut** [21] is not included in Table III because it estimates pixel locations instead of 3D positions. To compare the results, we projected the estimated and ground truth 3D positions into the input images and evaluated the pixel distance error. Experiment c in Table I showed a root-mean-square error at the tip position of 1.13 pixels in one camera view and 1.18 pixels in the other. In comparison, **DeepLabCut** achieved an accuracy similar to the human labeling error of 2.69 ± 0.1 pixels. However, comparing pixel errors is only of limited value, since a pixel error can have a different significance depending on the image resolution and scale of the captured object. Moreover, reprojecting the estimated pixels from multiple calibrated cameras back into 3D space may bring in additional errors due to camera calibrations. Therefore we believe that a direct 3D position estimation is more useful and convenient for downstream applications.

5) **Maximum Estimation Errors:** The maximum estimation errors are computed as an indication of the “worst case scenarios”. For WaxCast arm with features, the maximum tip error is 2.6% (VGG_s_bn), for SoPra, it is 4.2% (VGG13), and for the soft robotic fish, it is 5.3% (EffNet-B0).

6) **Online Estimation:** The online estimation performance of ViSE was tested on both a portable computer (2-core, 2.70 GHz Intel Core i7-7500U CPU, no GPU) and an Omen desktop computer (24-core, 3.2GHz Intel Core i9-
TABLE III
ESTIMATION ERRORS OF OUR APPROACH COMPARED TO OTHER WORKS.

| Technique            | Number of Cameras | Exact Contour Extraction | Good Contrast | Shape Agnostic | Robot Type     | Robot Length (mm) | Tip Error $^9$ | Estimation Freq. (Hz) |
|----------------------|-------------------|--------------------------|---------------|----------------|----------------|-------------------|----------------|----------------------|
| VISE (ours) VGG$_{2, bn}$ | 2                 | No                       | Required      | Yes            | WaxCast arm   | 335               | 0.3% ± 0.2%  | 1.6                  |
| VISE (ours) VGG13    | 2                 | No                       | Required      | Yes            | SoPrA         | 270               | 0.5% ± 0.4%  | 106                  |
| ViSe (ours) EffNetV2, m | 2               | No                       | Required      | Yes            | Soft arm      | 115               | 0.6% ± 0.6%  | 8.6                   |

Camarillo et al. [15] 2D point-cloud fit 3 Required Required No Soft arm 160 4.8% 3–4
Vandini et al. [18] Line feature detector 1 No Required No Soft arm 260 2.8% 0.1
Pedari et al. [37] LED light placement 2 No Required Yes Soft arm 287 4.5% ± 3.1% N/A
AlBeladi et al. [10] Edge detection & curve fit 1 Required Required No Soft arm 115 0.6% ± 0.6% 8.6

$^9$ Distance error normalized with the corresponding robot’s length. $^10$ Not provided, calculated based on their estimation data.

As reported in original works, not tested on the same machine, ViSe results are tested with a NVIDIA GeForce 3090 24 GiB GPU.

![Fig. 5. Performance of VGG13 on SoPrA under different experimental setups, with sample gray-scale images and processed masks shown. (a) Tip estimation errors with varying image brightness. The brightness modification is quantified by the addition or subtraction of pixel values (0–255) from original gray-scale images. (b) Tip estimation errors with increasing Gaussian noise. The noise with standard deviation from 0 to 50 is added per pixel to the pixel values of original gray-scale images. (c) Marker 1 and marker 2 (tip) estimation errors plotted with black strip occlusions of varying width at the marker positions.](image)

12900K CPU, 64GB memory, NVIDIA GeForce 3090 GPU with 24GB memory). A single estimation using VGG$_{2, bn}$ architecture for SoPrA takes 54 ms (18.4 Hz) on average on the portable computer, of which 60% are used for the CNN forward calculation and 38% are used for image processing. The remaining 2% are used to stream the images from the RGB cameras. The estimation rate can be greatly improved with the use of a GPU – CNN forward calculation and image processing on the desktop computer take only 1.96 ms and 6.77 ms, respectively, giving a theoretical estimation frequency of over 100 Hz. However, the real-world online estimation rate in this case is currently limited by the camera frequency of 30 Hz.

7) Different Experimental Setups: To demonstrate the robust performance of our proposed pipeline with adaptive thresholding in a range of lighting conditions without the need of retraining CNN, we evaluate and present the tip estimation errors on the SoPrA testing data set with modified brightness levels (see Figure 5a) and added Gaussian noise (Figure 5b). The brightness of the original images is modified by adding or subtracting pixel values from the grayscale images (pixel value range 0-255). The Gaussian noise is added per pixel to the original grayscale images with increasing standard deviation of the noise distribution. The experiments are conducted on the SoPrA data set with the best-performed VGG13 network since the gray color of the SoPrA arm is the closest to the black background. SoPrA provides the least contrast compared to the other soft robots and is, therefore, more sensitive to changes in brightness.

In the scope of this letter, we do not explicitly consider the problem of occlusion. Preliminary evaluation is carried out by inserting black strips of varying width at the marker position to test the inference robustness of the trained VGG13 network. The result is shown in Figure 5c.

The performance of the trained VGG13 network is also tested after the reassembly of the cameras. With relative camera translations and orientations obtained from the fiducial markers, we manually realign the cameras to a configuration with 1.46 mm difference from the one used during data collection. The new tip estimation error after reassembly of the cameras is 1.5 ± 1.6% for the SoPrA test data set.
C. Discussion

The results show that the estimation errors increase along the shape regardless of the data set or shape representation being used. This increase is most likely due to the fact that the tips of these shapes typically move faster and across a larger space than the rest of their shapes. A dynamic behavior increases the estimation difficulty towards the tip.

The approach using the PCC shape representation as output produces larger estimation errors on the three tested data sets (Table I). This error is partially because the endpoint positions are computed using forward kinematics calculation with all previous PCC sections, accumulating the estimation errors of each section. Another reason for the inferior performance of PCC is that the constant curvature assumption is sometimes inaccurate for a real soft robotic system. For example, the sections of the soft arm do not exactly bend with constant curvature. The arm’s weight and dynamics, the design characteristics of the inflation chambers, and the fabrication errors all introduce imperfections with regard to the constant curvature assumption. Moreover, the arms we use in this work do not contain an inextensible backbone and therefore also extend along their center line under inflation. This limitation could be resolved by augmenting the PCC model to allow for constant curvature sections of variable length. The CNN would then need to be adjusted to also estimate the length of each segment. However, the error due to non-constant curvature deformation would remain. By estimating points separately, we could avoid both the error accumulation and the PCC model limitations. Although the point representation does not contain any statements about connectedness or directionality, it gives a more precise estimation of the tip position.

The visual features added to the WaxCast arm greatly improved the estimation accuracy. This can be seen when comparing Experiments d and e in Table I. We believe that the added features helped the CNN extract more information from the input images. The increased information content improved the deduction of the shape parameters.

Performance of different CNN architectures is presented in Table II. Overall, VGG architectures exhibit the fastest estimation speed and the least tendency to overfit the training data set. Although the performance of VGG with batch normalization greatly decreased, we can see that the batch normalization helps with preventing overfitting. For the larger data sets of the WaxCast and SoPrA arms, VGG and EfficientNetV2 architectures do not overfit the training sets. However, due to the limited data set size for the soft fish, most CNN architectures (other than VGG with batch normalization and EffNetV2-m) tend to overfit for this training set. This overfitting is also because more recent CNN architectures, especially EfficientNet and EfficientNetV2 are designed to scale up training with larger image sizes, therefore they might overfit small data sets more easily and cannot outperform simpler VGG architectures on tasks with small binary image input. Among the architectures that do not overfit on the training data of the soft fish, EffNetV2-m performs the best in terms of the mean estimation error on the testing set. In practice, the EffNetV2-m architecture is least favored for our application since its average online estimation frequency is below 10 Hz.

We outperform the benchmarks for both soft arms and achieve slight performance improvements for the soft fish, as shown in Table III. At the same time, our approach also does not require to extract contour lines or any prior knowledge of the shape, suggesting the possible generalizability to different types of soft robots.

One limitation of using convolutional neural networks is that the trained network may not be reusable and needs retraining when the experimental setup changes. We tested our trained network on input images with various levels of brightness, noises, occlusion, and after reassembly of the cameras. The stable performance under brightness changes and gaussian noises (Figure 5a and b) indicates that the proposed method could work with a wide range of lighting conditions without re-training the CNN as long as there is sufficient contrast for the adaptive image preprocessing. Since we do not consider occlusions during the training phase, the estimation performance decreases with increasing occlusions during inference. However, compared to marker-based method, the CNN is capable of predicting marker positions even with the markers fully occluded. When occluding marker 2 up to a width of 20 pixels, the estimated position error of marker 1 only slightly decreases by 0.63% ± 0.39% compared to the unobstructed case (Figure 5c). This shows the potential robustness of the proposed method against occlusion. Although retraining would be needed for different camera configurations, we show with the aid of fiducial markers (AprilTags), rough realignment to previous camera positions is possible and the trained network can be reused.

V. CONCLUSIONS AND FUTURE WORK

VISe is a vision-based, 3D soft robot shape estimation approach using two cameras and a CNN. It outperforms current marker-less shape estimation approaches when evaluated on two soft robotic arms and one soft robotic fish. While we consider the visual robustness of our approach to be an improvement over the state-of-the-art, it could be further enhanced to be calibration-free, deal with occlusions, and allow for more expressive representations. Future work will introduce artificial occlusions in the network’s training process to work with partially occluded images and also employ learning-based shape segmentation to perform robust background removal under insufficient contrast. Another future direction is to generalize the approach to the estimation of more expressive kinds of shape representations, e.g., mesh reconstructions, instead of being limited to the estimation of piecewise constant curvatures or characteristic points.

REFERENCES

[1] Y. Yamanaka, S. Katagiri, H. Nabae, K. Suzumori, and G. Endo, “Development of a food handling soft robot hand considering a high-speed pick-and-place task,” in IEEE/SICE Int. Symp. Syst. Integration, 2020, pp. 87–92.

[2] Z. Wang, K. Or, and S. Hirai, “A dual-mode soft gripper for food packaging,” Robot. Auton. Syst., vol. 125, p. 103427, 2020.

[3] E. W. Hawkes, C. Majidi, and M. T. Tolley, “Hard questions for soft robotics,” Sci. Robot., vol. 6, no. 53, p. eabg6049, 2021.

[4] O. Yasa, Y. Toshimitsu, M. Y. Michelis, L. Jones, M. Filippi, T. Buchner, and R. K. Katzschmann, “An Overview of Soft Robotics,” Annu. Rev. - Cont. Robot. Auton. Syst., 2022.
