Vision-Based Online Key Point Estimation of Deformable Robots

Journal Article

Author(s): Zheng, Hehui; Pinzello, Sebastian; Cangan, Barnabas Gavin; Buchner, Thomas J.K.; Katzschmann, Robert K.

Publication date: 2024

Permanent link: https://doi.org/10.3929/ethz-b-000682228

Rights / license: Creative Commons Attribution 4.0 International

Vision-Based Online Key Point Estimation of Deformable Robots

H. Zheng,* Sebastian Pinzello, Barnabas Gavin Cangan, Thomas J. K. Buchner, and Robert K. Katzschmann*

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, a regression approach is presented for three-dimensional key point estimation using a convolutional neural network. The proposed approach uses data-driven supervised learning and is capable of online markerless estimation during inference. Two images of a robotic system are captured simultaneously at 25 Hz from different perspectives and fed to the network, which returns for each pair the parameterized key point or piecewise constant curvature shape representations. The proposed approach outperforms markerless state-of-the-art methods by a maximum of 4.5% in estimation accuracy while being more robust and requiring no prior knowledge of the shape. Online evaluations on two types of soft robotic arms and a soft robotic fish demonstrate the method’s accuracy and versatility on highly deformable systems.

1. 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 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 and increased fabrication complexity. This limitation has lately led to an increased popularity of exteroceptive sensing approaches that are purely vision based.[9,10]

A widely applied vision-based method for key point estimation and tracking employs motion-capture systems.[9,11,12] These systems rely on placing reflective markers on an object and triangulating the marker position using multiple calibrated motion-tracking cameras. Such marker-based approaches offer high temporal and spatial resolutions of the placed markers. However, these tracking systems are usually costly and limited in use, hindering the application of marker-based robots in commercial settings where multiple robots are deployed simultaneously within clutter.[13] Motion-capture systems require the tedious placement of many markers on any object to be tracked, which is not a feasible option when working with high-dimensional robotic systems that interact with an environment. Markers cannot be placed too close to each other, limiting the fidelity of shape reconstruction. Furthermore, obstacles could occlude and eventually displace or destroy the markers. Therefore, much research has gone into replacing marker-based approaches by developing markerless sensing techniques.[8,10,14-18] Current markerless approaches still require scenario-specific capturing setups and strict formatting requirements of the image data.
In this work, we choose an approach toward 3D key point estimation of continuously deformable robots that leverage data-driven deep learning. We propose VOKE, a Vision-based Online 3D key point estimation approach for continuously deformable robots that leverage data-driven deep learning. We propose VOKE, a Vision-based estimation of continuously deformable robots that leverage data-driven deep learning. We propose VOKE, a Vision-based estimation of continuously deformable robots that leverage data-driven deep learning. We propose VOKE, a Vision-based estimation of continuously deformable robots that leverage data-driven deep learning. We propose VOKE, a Vision-based estimation of continuously deformable robots that leverage data-driven deep learning.

Previous works demonstrate vision-based shape parameter estimation approaches for continuously deformable robots that either work only for specific setups or necessitate strict image requirements, for example, exact segmentation for contour extraction. In the following, we briefly discuss the most related works, focusing on markerless 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. However, their estimation results are only compared to another cable-based shape estimation technique but not with actual ground truth. Camarillo et al. extended the computer vision methods to 3D spline estimation of a thin continuum manipulator. If the precise positions of cameras are known, they could extract silhouettes from multiple camera's 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. Croom et al. also performed edge detection but then fit reference points to the edges using an unsupervised learning algorithm called the self-organizing map. All of these approaches show good estimation results but require a strong contrast between the tracked object and the background.

Vandini et al. extracted and joined straight lines from a mono-plane fluoroscopic surgical image to estimate the shape of a concentric tube robot. By posing conditions for connecting the line features, they managed to relax image requirements and can extract curves from more unclean image data compared to the aforementioned works. Reiter et al. took on a similar approach to ours in that they extracted 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 markerless pose estimation and tracking called DeepLabCut. 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 key point estimation of continuously deformable robots employs CNNs. While there are many vision-based proprioceptive methods for soft robots using deep learning, we focus on exteroceptive approaches that are simple to implement and do not add complexity to the manufacturing of the soft robots.

2. Related Works

Figure 1. Diagram of the markerless online inference pipeline for the proposed key point estimation approach VOKE. A 3D shape is captured by two RGB video cameras. Image pairs are preprocessed and run through a CNN to estimate a shape model for the 3D shape.

Previous works demonstrate vision-based shape parameter estimation approaches for continuously deformable robots that

either work only for specific setups or necessitate strict image requirements, for example, exact segmentation for contour extraction. In the following, we briefly discuss the most related works, focusing on markerless 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. However, their estimation results are only compared to another cable-based shape estimation technique but not with actual ground truth. Camarillo et al. extended the computer vision methods to 3D spline estimation of a thin continuum manipulator. If the precise positions of cameras are known, they could extract silhouettes from multiple camera's 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. Croom et al. also performed edge detection but then fit reference points to the edges using an unsupervised learning algorithm called the self-organizing map. All of these approaches show good estimation results but require a strong contrast between the tracked object and the background.

Vandini et al. extracted and joined straight lines from a mono-plane fluoroscopic surgical image to estimate the shape of a concentric tube robot. By posing conditions for connecting the line features, they managed to relax image requirements and can extract curves from more unclean image data compared to the aforementioned works. Reiter et al. took on a similar approach to ours in that they extracted 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 markerless pose estimation and tracking called DeepLabCut. 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 key point estimation of continuously deformable robots employs CNNs. While there are many vision-based proprioceptive methods for soft robots using deep learning, we focus on exteroceptive approaches that are simple to implement and do not add complexity to the manufacturing of the soft robots.

3. Methods

Our proposed estimation method is a learnt multiview parametric 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 CNN trained to estimate the parameters of the shape representation. The approach is outlined in Figure 1 and the following subsections detail the subcomponents of our method.

3.1. Image Preprocessing

The red, green and blue (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.

3.2. Shape Representations

We consider two different parametric shape representations, the point position and PCC model. Ground truth shape is obtained using the motion capture software (Qualisys 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 fit 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, six instead of nine for a three-section soft arm, while representing a long, continuously deformable shape, which is useful for model-based control purposes.

3.3. 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, EffNet-B1, EffNet-B3, EffNet-B5), as well as 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 1.

3.4. Camera Realignment

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, rectified linear units (ReLU), 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 2, 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 8,192 (8 × 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 1,000, which is reduced to the output size of 6, 9, 12, or 18 (Table 1).

Table 1. Estimation errors of the PCC and point estimation approaches.

<table>
<thead>
<tr>
<th>Exp.</th>
<th>Repr.-Sec.</th>
<th>Out.</th>
<th>a) Feat.</th>
<th>b) Point 1 [%]</th>
<th>c) Point 2 [%]</th>
<th>d) Point 3 (tip) [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>PCC-6</td>
<td>12</td>
<td>Yes</td>
<td>0.9 ± 0.3</td>
<td>3.2 ± 1.4</td>
<td>6.1 ± 2.4</td>
</tr>
<tr>
<td>b</td>
<td>PCC-3</td>
<td>6</td>
<td>No</td>
<td>1.3 ± 0.6</td>
<td>2.6 ± 1.3</td>
<td>6.8 ± 3.5</td>
</tr>
<tr>
<td>c</td>
<td>Point-6</td>
<td>18</td>
<td>Yes</td>
<td>0.06 ± 0.03</td>
<td>0.1 ± 0.07</td>
<td>0.4 ± 0.3</td>
</tr>
<tr>
<td>d</td>
<td>Point-3</td>
<td>9</td>
<td>No</td>
<td>0.4 ± 0.4</td>
<td>0.8 ± 1.1</td>
<td>3.6 ± 5.0</td>
</tr>
<tr>
<td>e</td>
<td>Point-3</td>
<td>9</td>
<td>Yes</td>
<td>0.06 ± 0.03</td>
<td>0.1 ± 0.06</td>
<td>0.3 ± 0.2</td>
</tr>
</tbody>
</table>

a) CNN output size. b) Visual features. c) Point positions in Figure 3, distance error normalized with the robot’s length (335 ± 3 mm).

Figure 2. Truncated VGG network architecture: VGG-s-bn. Inputs are the preprocessed binary images from both cameras, and output sizes depend on the selected shape representation and robot (Table 1).
markers (AprilTags) are attached to the robot during data collection. To alleviate the need for retraining, fiducial markers are fixed relative 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 user can realign their cameras to perform inference using the original training data. Given our realignment utility, the cameras can be reset to the approximate same relative poses to the robot's base; then the trained model can successfully estimate the key point positions of the robot without requiring a motion-capture system to retrain.

4. Results

Our approach is tested on two types of soft robotic arms [11, 33] and a soft robotic fish [34] (Figure 3). The detailed description of the soft robots can be found in the Experimental Section.

4.1. Parametric Shape Representations

The CNN was trained using the image data and either learnt 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 datasets from our two WaxCast arms (Figure 4a–d). Detailed results of the evaluation using VGG-s-bn can be seen in Table 1 for Experiment 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 ± 3 mm for the soft arm.

4.2. Visual Features

To evaluate the effect of visual features on the estimation accuracy, the point estimation approach is applied to a dataset 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 1, Experiment 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.

4.3. CNN Architectures

We tested 14 different CNN architectures and reported their performance on 3 soft robots in Table 2. All errors are normalized with respect to the robot's length. The main performance metrics are mean and maximum tip estimation errors of the testing dataset. Considering the limited data size, we prefer networks with better generalizability outside of the training dataset. Therefore, we also include the overfit ratio, which is calculated by dividing the mean tip estimation error of the testing data set by the training mean errors. To avoid possible overfitting to the training set, for each dataset, 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 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 NVIDIA Tesla V100-SXM2 32 GiB graphics processing unit (GPU).

4.4. Maximum Estimation Errors

The maximum estimation errors are computed as an indication of the “worst case scenarios.” For WaxCast arm with features, the Figure 3. Soft robots used for performance evaluation with evaluation point positions illustrated; in each panel, left shows the original RGB image, right shows the preprocessed image. a) WaxCast Arm,[11] b) WaxCast arm with visual features (black stripes),[11] c) SoPrA arm,[33] and d) Soft fish.[34]
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).

### 4.5. Benchmarks

We compared the results of our point estimation approach with four similar works (see Table 3), which also estimated reference points along continuously deforming shapes. Since the previous works either are shape agnostic or use active light-emitting diode markers, they are unsuitable for our application with different soft robots. Due to limited data and the lack of open-source code/platform to reproduce the benchmarks, especially for the concentric robots for medical use, 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 is not included in Table 3 because it estimates pixel locations instead of 3D positions.\textsuperscript{23} 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 3 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.

4.6. Online Estimation

The online estimation performance of VOKE 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, 3.2 GHz Intel Core i9-12900 K CPU, 64 GB memory, NVIDIA GeForce 3090 GPU with 24 GB memory). A single estimation using our truncated VGG-s-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.

4.7. 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 dataset 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 dataset 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 work, 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 dataset.

4.8. 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 toward the tip.

The approach using the PCC shape representation as output produces larger estimation errors on the three tested datasets (Table 1). 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.

Table 3. Estimation errors of our approach compared to other works.

<table>
<thead>
<tr>
<th>Estimation technique</th>
<th>Number of cameras</th>
<th>Require exact contour</th>
<th>Shape agnostic</th>
<th>Robot type</th>
<th>Robot length [mm]</th>
<th>Tip Error\textsuperscript{a} [%]</th>
<th>Freq\textsuperscript{b} [Hz]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (VGG-s-bn)</td>
<td>CNN</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
<td>335</td>
<td>0.3 ± 0.2</td>
<td>146</td>
</tr>
<tr>
<td>Ours (VGG13)</td>
<td>CNN</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
<td>270</td>
<td>0.5 ± 0.4</td>
<td>106</td>
</tr>
<tr>
<td>Ours (EffNetV2-m)</td>
<td>CNN</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
<td>115</td>
<td>0.6 ± 0.6</td>
<td>8.6</td>
</tr>
<tr>
<td>Camanillo et al.\textsuperscript{11}</td>
<td>2D point-cloud fit</td>
<td>3</td>
<td>Yes</td>
<td>No</td>
<td>160</td>
<td>4.8</td>
<td>3–4</td>
</tr>
<tr>
<td>Vandi et al.\textsuperscript{19}</td>
<td>Line feature detector</td>
<td>1</td>
<td>No</td>
<td>No</td>
<td>260</td>
<td>2.8</td>
<td>0.1</td>
</tr>
<tr>
<td>Pedari et al.\textsuperscript{35}</td>
<td>LED light placement</td>
<td>2</td>
<td>Yes</td>
<td>Yes</td>
<td>468\textsuperscript{4}</td>
<td>4.5</td>
<td>N/A</td>
</tr>
<tr>
<td>AlBeladi et al.\textsuperscript{10}</td>
<td>Edge detector &amp; curve fit</td>
<td>1</td>
<td>Yes</td>
<td>No</td>
<td>287</td>
<td>4.5 ± 3.1</td>
<td>N/A</td>
</tr>
</tbody>
</table>

\textsuperscript{a}Error normalized with the corresponding robot’s length; \textsuperscript{b}Estimation frequency as reported in original works, not tested on the same machine, our results are tested with a NVIDIA GeForce 3090 24 GiB GPU; \textsuperscript{c}Not provided, calculated based on their estimation data.
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 nonconstant 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 1. 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.

The performance of different CNN architectures is presented in Table 2. 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 datasets of the WaxCast and SoPrA arms, VGG and EfficientNetV2 architectures do not overfit the training sets. However, due to the limited dataset size for the soft fish, most CNN architectures (other than VGG with batch normalization and EfficientNetV2-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 datasets more easily and cannot outperform simpler VGG architectures on tasks with small binary image input. Among the architectures that do not overfit 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 all three soft robots, as shown in Table 3. 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 CNNs 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,b) indicates that the proposed method could work with a wide range of lighting conditions without retraining 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 unoccluded 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.

Figure 5. Performance of VGG13 on SoPrA under different experimental setups, with sample grayscale 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 grayscale 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 grayscale images. c) Marker 1 and marker 2 (tip) estimation errors plotted with black strip occlusions of varying width at the marker positions.
5. Conclusion

VOKE is a vision-based, 3D soft robot key point estimation approach using two cameras and a CNN. It outperforms current markerless 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 use 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, for example, mesh reconstructions, instead of being limited to the estimation of PCC or characteristic points.

6. Experimental Section

Soft Robots for Evaluation: The approach was tested on two types of soft robotic arms and a soft robotic arm (Figure 3). The first soft robotic arm (which we shall refer to as the WaxCast arm) consisted of three axially connected cylindrical segments, each with four separately inflatable chambers. They were inflated using air provided through a pressure-controlled valve array (Festo SE & Co. KG). By inflating one side, the chambers on that side elongated and induced bending in the segment; thus, the bending direction of the arm can be chosen by selecting the corresponding combination of inflation chambers. Each segment had a length of about 110 ± 1 mm and a diameter of 40 ± 1 mm. The second arm, SoPrA, is a two-segment soft robotic arm with fiber-reinforced pneumatic actuators. Segments were made of three individually fiber-reinforced elastomer air chambers that were glued together. Combining two of these segments added up to a total length of 270 ± 2 mm. The robotic fish tail was similar in construction and actuation compared to the WaxCast arm, except that it was shaped like a fishtail. It consisted of two inflatable chambers and had a total length of 115 ± 1 mm.

Data Collection: The ground truth data for learning was obtained by eight Miqus M3 motion-capture cameras from Qualisys AB placed in the motion-capture space of 1.6 × 1.1 × 0.8 m. The placement of the motion-capture markers is shown in Figure 3. A group of reflective markers was placed on a rigid ring at the end of every segment of the soft arm. Along the soft fish’s tail, the markers were placed with spaces of 38 ± 1 mm between them. Marker position data was supplied at 100 Hz with an average accuracy of 0.1 ± 0.2 mm, while RGB image data was recorded at 25 Hz by two depth cameras (Intel RealSense D435i).

During the data acquisition, the pose data (i.e., the specific shape configurations of the soft robots) was collected such as to cover the robots’ full workspace. The segments of the WaxCast arm were all actuated to perform a circular motion, with periods of 100, 10, and 1 s, for segments 1, 2, and 3, respectively. We created two datasets, one with three motion-capture marker rings on the arm and the other with six, which also contained 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 were generated for two arms, each containing 12,000 poses. The soft robotic fish was actuated to perform a tailfin stroke with maximal deflection, resulting in a dataset of 1800 poses. Each dataset was split into 90% training and 10% testing sets.

Network Training: The network was implemented and trained using the PyTorch framework. AdamW was chosen as an optimizer and used to minimize the mean absolute loss. The network was trained per robot using a batch size of 64 with an early stop for a maximum of 450 epochs on each dataset. The learning rate was set to 10^-4 and reduced by 0.5 after each 200th epoch. Dropout was 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 and 60 min of convection.

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This research at the Soft Robotics Lab was generously funded by Credit Suisse, Switzerland. A doctoral fellowship by the ETH AI Center partially supports Hehui Zheng and a Swiss Government Excellence Scholarship partially supports Barnabas Gavin Cangan. This project is part of Swiss Data Science Center Collaborative Project C22-08.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability Statement

The data that support the findings of this study are openly available in ETH Research Collection at https://doi.org/10.3929/ethz-b-000678014, ref. [38]. The full pipeline is open-sourced on https://github.com/srl-ethz/voke, ref. [39].

Keywords

key point estimations, soft robots, vision-based estimations

Received: February 6, 2024
Revised: May 27, 2024
Published online:
