

# A deep neural network approach for accurate 3D shape estimation of soft manipulator with vision correction

Shuangquan Zou, Yueyong Lyu <sup>\*</sup>, Jiaming Qi, Guangfu Ma, Yanning Guo

*Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China*



## ARTICLE INFO

### Keywords:

Soft manipulator  
3D shape estimation  
Deep neural network  
Constant curvature kinematics  
Vision sensing

## ABSTRACT

Soft manipulator is a strong non-linear and uncertain system with infinite degrees of freedom. The real-time 3D shape estimation is the guarantee for control and application, but a single kind of sensor technology always has inherent limitations. To address the above issue, firstly, the constant curvature (CC) kinematics is proposed to roughly estimate the 3D shape of the soft manipulator, because this method becomes inaccurate when the manipulator is bent with large deformation. Secondly, a vision-based high-precision shape estimation method is developed. The self-organizing map (SOM) adaptive algorithm is introduced to identify the centerline of the manipulator from the image data. While vision sensing has strict requirements for the environment. Therefore, a learning approach based on a deep neural network (DNN) is designed to correct the CC kinematics with accurate visual estimation results. Finally, the performance of the trained DNN is evaluated on the test set and a real-time bending deformation experiment. The results indicate that the DNN approach has high accuracy and stability ability to learn the entire 3D shape of the soft manipulator in real time.

## 1. Introduction

Soft manipulator [1] is an emerging research field in robotics and have therefore been extensively studied. In recent years, more and more research on the soft manipulator has gradually developed from theoretical simulation to assessment experiment and even practical application. Especially in unstructured and constrained environments such as ground rescue, industrial inspection, and unknown exploration [2], the natural dexterity and inherent compliance allow the soft manipulator to safely adapt to narrow paths by changing its shape, which is an impossible task for the traditional rigid manipulator.

However, the flexibility and nonlinear driving deformation brought by flexible materials often hinder the accurate control of the soft manipulator in applications. In complex and constrained environments, real-time shape information is essential to improve control accuracy, and visualizing the shape of the soft manipulator also facilitates users to better operate and avoid obstacles. To date, numerous classical and effective shape sensing methods have been proposed and applied to practical systems [3,4].

In this application scenario, internal sensors embedded or integrated within the manipulator become a better choice. A common approach to estimating the shape of the soft manipulator is measuring the length of

the pneumatic muscle actuators (PMAs) with a cable encoder as a sensor to develop the constant curvature (CC) kinematics model [5], and the hardware experiments have been implemented on a variety of multi-section soft manipulator. But this method overly relies on the CC assumption so that is not applicable in the case of large deformation [6] or gravity load effect [7]. Thus, some studies have emerged to moderately modify the CC kinematics. Godage et al. [8] applied the modal kinematics model to circumvent the singular problem in CC kinematics but could not detect the non-constant curvature deformation yet. A variable curvature modeling method based on Euler spirals [9] is addressed to match the real 2D shape better in some poses. Dynamic model can more accurately model the shape of the soft manipulator. Rone [10] adopted length variables to deduce the dynamic model through the virtual power principle to capture the curvature change and 3D torsion along the backbone. Almost all dynamic models [11,12] consider complex mechanical models with massive calculations which increases the difficulty of real-time control, and the placement of the force sensor also affects the flexibility of the soft manipulator. Regarding other types of sensors, Song et al. [13] installed electromagnetic sensors at the distal end of a wire-driven flexible manipulator to track shape information in real time. Fiber optic sensor for continuously changing media has been experimentally demonstrated to capture the actuator's

\* Corresponding author.

E-mail address: [lvyy@hit.edu.cn](mailto:lvyy@hit.edu.cn) (Y. Lyu).

shape and tip position with submillimeter resolution [14]. The above-mentioned solutions based on internal sensors all require mathematical analysis models, whereas there are many inexplicable characteristics in the modeling process, including parameter uncertainty, friction, and hysteresis of soft materials. For convenience, many ideal assumptions are defaulted, resulting in the shape sensing tending to be imprecise for this type of sensor.

On the contrary, some solutions use the relatively reliable vision-based external sensor, as this approach [15] only requires image data to accomplish 3D shape recognition with the highest possible accuracy. From the pioneering work of configuring the monocular camera to monitor the plane motion of the elephant-trunk robot [16], to later establishing stereo [17] and trinocular [18] vision systems, scholars are trying to deal with the challenge of shape estimation of the soft manipulator by machine vision, and some cases have been considerable advances in minimally invasive surgery [19]. But visual sensing is not feasible to directly apply in outdoors or constrained environments.

As we can see, considerable research efforts have been devoted to making various improvements to the shape sensing system with a single kind of sensor, whether in a physical model or sensing technology. However, in essence, it is still unable to break the inherent defects brought by the single kind of sensor. The learning-based approaches [20], namely machine learning (ML), have been fully identified as an effective solution to improve the scope. For example, Loutfi et al. [21] compared the performance of four different learning models in the forward kinematics of the continuum manipulator. The supervised learning method and the Jacobian method for solving the inverse statics of a cable-driven soft arm with non-constant curvature are discussed in [22]. Melingui et al. [23] designed an adaptive neural network controller that enables the compact bionic handling arm to track the desired path in real time with high accuracy. Gaussian mixture model [24] is reported to learn multiple trajectory models of demonstration teaching in the surgical task. With the development of GPU computing power and open access packages, deep learning provides more advanced alternatives to learn and optimize the results of ML. Wang et al. [25] took advantage of the convolutional neural network to encode the image into the deformation state and trained a decoder neural network to reconstruct the soft body's 3D shape. Li et al. introduced the long-short-term-memory neural network to dynamically estimate the distal force of the flexible endoscopic robot [26]. Su et al. [27] constructed an improved recurrent neural network-based control scheme to optimize the end error of the surgical robot within 4 mm. Also, the incremental learning process has been proved to optimize redundancy and achieve fast and efficient learning [28]. Little is done, still, about the study of shape estimation with learning approaches. There are merely some studies on learning the real end position of the soft manipulator followed by multilayer perceptron (MLP) [29,30]. Unlike the rigid-link manipulator, the end position is not a good measure of the complete shape of the soft manipulator since this continuous deformation could be in any one state of infinite possibilities. Compared with MLP, deep neural network (DNN) emphasizes the depth of the model structure, that is, it has more hidden layers and enough hidden neurons to learn more features in essence. Hence, perhaps it can be utilized to learn all information about the 3D shape of the soft manipulator, not just the position of the end.

In this work, a DNN approach is presented to establish the mapping relationship between length information and visual markers to estimate the 3D shape of the soft manipulator accurately. Our main contributions are outlined as follows:

- (1) The advantages and limitations of constant curvature kinematics and visual shape estimation are analyzed in detail through experimental simulation comparisons.
- (2) A DNN approach is raised to learn the entire 3D shape of the soft manipulator, which shows great stability and accuracy in the test set and real-time deformation experiments.

- (3) The proposed learning-based method circumvents the complicated modeling process, and ingeniously complements the advantages of two sensing systems. This general approach is instructive and can be applied to any continuum manipulator.

## 2. System overview

In the feedback loop, the quality of the shape estimation system attaches great importance to the whole closed-loop control system. A soft manipulator platform is set up to conduct the relevant experimental support verification (see Fig. 1), containing electric control system, pressure-actuated system, and shape estimation system.

The 24 V DC power supply and self-made buck module meet the various voltage requirements of the whole platform. The air compressor and valve island (Festo, Germany) provide the high-pressure source, and the DA converter converts the control signal into  $U^C$  to control  $P^C$ , and finally the shape of the soft manipulator is deformed by means of pressure drive.

There are two shape estimation systems in the experimental setup (discussed in Sections 3 and 4). The microcontroller STM32 reads the signals from the cable encoders and wirelessly transmits  $\{l_{11}, l_{12}, l_{13}\}$  to the workstation via Bluetooth. The ASES outputs  $\hat{Q}$  by the stereo camera (ZED, USA), but it is just offline training the DNN model in advance. After the DNN model is obtained,  $\hat{Q}$  can be output only by running the RSES in real time, which implements vision correction (more details in Section 5.1). The simulation of the 3D shape for the soft manipulator can be displayed on the workstation.

### 2.1. Prototype description

We designed and fabricated a pressure-actuated soft manipulator for the unit section in this work. The component unit of the prototype is the PMA, made by covering the surface of the silicone rubber tube with the nylon mesh. a pneumatic connector is directly plugged into the silicone rubber tube to control the flow of air, while the other end is tightly sealed with the help of the cable tie (Fig. 2). Each PMA has an individual chamber as the pressure control channel, which can extend up to 200% and withstand 0–3 bar pressure range. Three identical PMAs, distributed symmetrically in parallel, are circumferentially connected to actuate the prototype together. Besides, the gaps between PMAs are tightly fixed by the braided thread to ensure the prototype's shape is as close as possible to the feature of CC. Compared with other prototypes [31], the actuator configuration we designed can bend in any direction instead of one when each PMA is filled with different pressure.

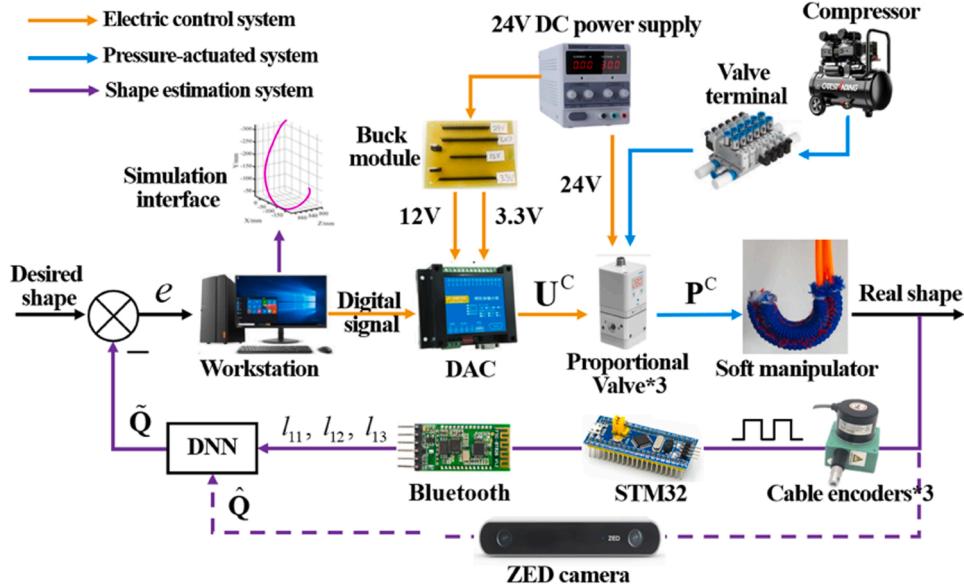
### 2.2. Sensor network distribution

As the sensor distribution of RSES, three cable encoders are installed  $120^\circ$  apart from each other at a radius  $d$  from the center of the circular base. Note that the cables from encoders parallel to the centerline are attached in the gap between PMAs and consistent with the length variation of PMAs for accurate measurement in real time. Here, we delivered a 3D illustration and top view of the sensor network distribution, as given in Fig. 3.

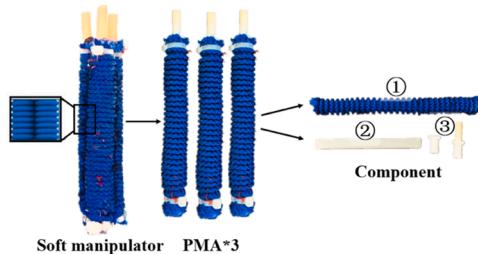
The visual shape detection system is quite simple with only ZED. It is fixed on a tripod with adjustable height to ensure all the workspace of the manipulator is covered by the vision field. All programs in this work are run on the workstation (Intel Xeon Bronze 3104 CPU, 32 GB RAM, NVIDIA Quadro P4000).

## 3. RSES based on CC kinematics

In this section, the CC kinematics based on PAM length is proposed to roughly estimate the 3D shape of the soft manipulator. The experiment results then illustrate the advantages and limitations of CC kinematics



**Fig. 1.** The block diagram of the whole soft manipulator platform.  $P^C \in [0, 10]\text{bar}$  is the pressure vector of three proportional valves (Festo, Germany).  $U^C$  is analog voltage vector as three outputs of the DAC.  $\hat{Q}$  denotes the results of the ASES, and  $\tilde{Q}$  is the prediction output of the DNN. Also, the dashed arrow indicates an offline system.



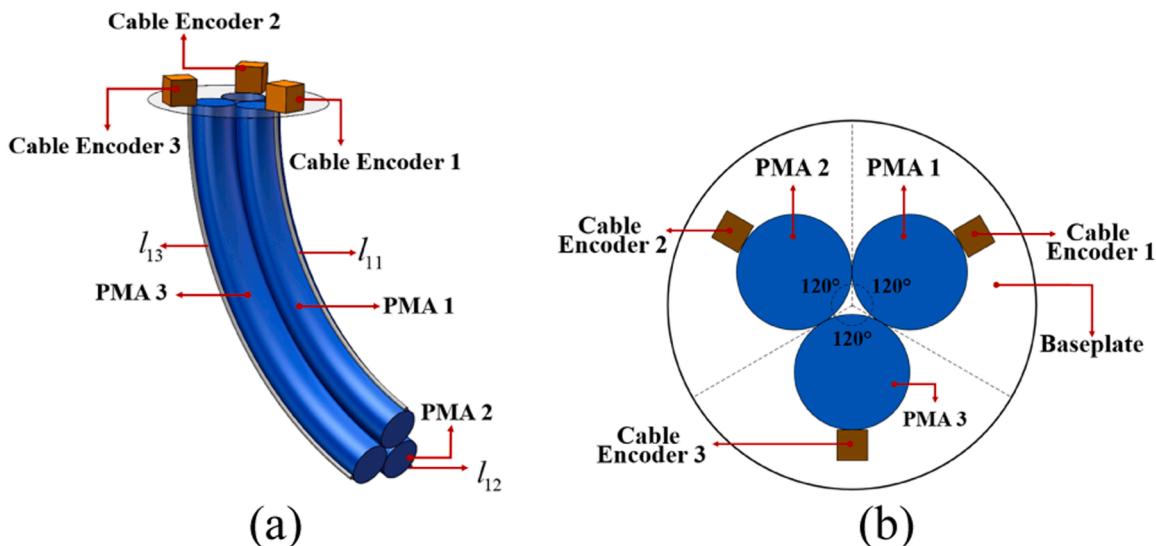
**Fig. 2.** Materials and local details of the prototype. ① is the nylon mesh to limit radial contraction and axial elongation for PMAs; ② is silicone rubber tube; ③ are pneumatic connectors made of 3D printing.

with several examples.

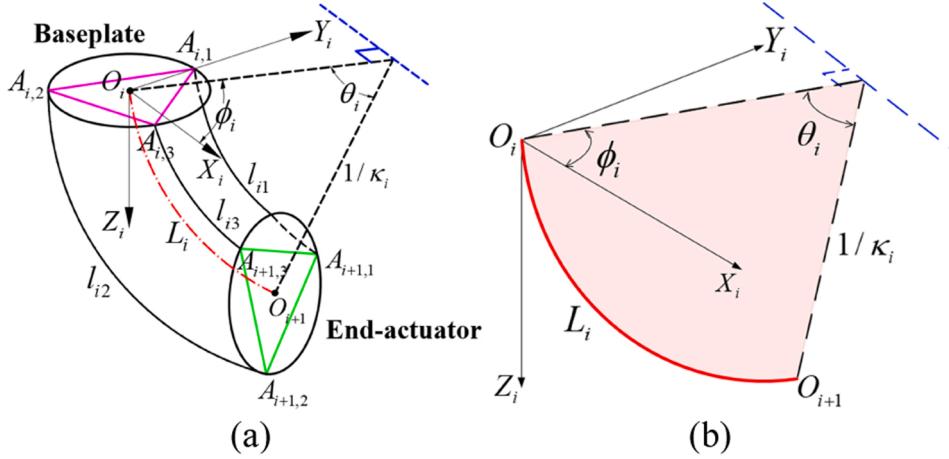
### 3.1. Model definition

The principle of the CC method indicates that the unit-segment soft manipulator bends as a standard arc, which simplifies the kinematics into an ideal characteristic. Therefore, some assumptions should be considered during the modeling process: (1) The prototype deforms continuously and evenly, and the bending sections of PMAs are equal in each segment; (2) Torsion can be ignored.

To illustrate the CC model, the unit-segment soft manipulator is designed in Fig. 4a.  $A_{ij}$  represents the position of the cable encoders, where  $i$  is the segment number, and  $j \in \{1, 2, 3\}$  refers to the cable encoders number. The length of the PMAs  $l_{ij} \in \mathbb{R}$  is parallel to the backbone  $O_iO_{i+1}$ , which is measured by cable encoders. Due to the material properties of our prototype, the variation range of the  $l_{ij}$  can reach  $l_{\min} = 77\text{mm}$  and  $l_{\max} = 252\text{mm}$  separately. We denote  $\mathbf{q}_i = [l_{i1}, l_{i2}, l_{i3}]^T \in$



**Fig. 3.** Sensor network distribution. (a) 3D illustration (b) Top view.



**Fig. 4.** Schematic of unit-segment soft manipulator. (a) The unit segment model. (b) Geometric schematic.

$\mathbb{R}^3$  as the joint vector for convenience. The local coordinate  $\{O_i - X_i Y_i Z_i\}$  is established in this way: With the center of the baseplate as the origin  $O_i$ , the  $Z_i$  axis is downward perpendicular to the baseplate. Besides, the  $Y_i$  axis passes through the point  $A_{i,1}$ .

The backbone  $O_i O_{i+1}$  depicts the spatial shape of the soft manipulator. It can be expressed by three spatial arc parameters  $\{L_i, \kappa_i, \phi_i\}$ .  $L_i$  is the arc length,  $\kappa_i \in (0, \infty)$  is specified as the curvature, and  $\phi_i \in [-\pi/2, 3/2\pi]$ , the bending plane angle, is defined as the angle formed by the  $X_i$  axis and the plane where the backbone is located. More details are given in Fig. 4b.

### 3.2. CC kinematics

Generally, CC kinematics is decomposed into two continuous processes to complete the mapping from joint space to workspace. (1) Joint space to configuration space:  $\mathbf{q}_i$  is first transformed into spatial arc parameters  $\{L_i, \kappa_i, \phi_i\}$  through the spatial geometric derivation  $F$ . This mapping is related to the structural design of the robot; (2) Configuration space to work space:  $\{L_i, \kappa_i, \phi_i\}$  are then transformed into the pose of end-actuator  $\mathbf{P}_i \in \mathbb{R}^3$  by improved D-H method. This mapping is universal and applicable to all constant curvature robots. The above mapping relationship can be concluded as

$$\mathbf{q}_i \xrightarrow{F} \{L_i, \phi_i, \kappa_i\} \xrightarrow{\text{D-H}} \mathbf{P}_i \quad (1)$$

The following paragraphs clarify the two mapping processes. Firstly, by means of spatial geometric derivation (refer to [32]), the spatial arc parameters in configuration space can be obtained with the help of joint vector

$$L_i(\mathbf{q}_i) = (l_{i1} + l_{i2} + l_{i3})/3 \quad (2)$$

$$\kappa_i(\mathbf{q}_i) = \frac{2\sqrt{l_{i1}^2 + l_{i2}^2 + l_{i3}^2 - l_{i1}l_{i2} - l_{i1}l_{i3} - l_{i2}l_{i3}}}{d(l_{i1} + l_{i2} + l_{i3})} \quad (3)$$

$$\phi_i(\mathbf{q}_i) = \tan^{-1} \left( \sqrt{3} \left( l_{i2} + l_{i3} - 2l_{i1} \right) / 3(l_{i2} - l_{i3}) \right) \quad (4)$$

where  $d$  is the radius of the baseplate. Note that formula (4) is further explained here. Since the range of  $\tan^{-1}(x)$  is  $[-\pi/2, \pi/2]$ , the pose of the soft manipulator cannot cover the entire 3D space. Considering the distribution of PMAs, we improve formula (4) according to the relative relation between the value of  $l_{ij}$

$$\phi_i^*(\mathbf{q}_i) = \begin{cases} \phi_i, (\phi_i > 0 \cap l_{i1} < l_{i2}) \cup (\phi_i < 0 \cap l_{i1} > l_{i3}) \\ \phi_i + \pi, (\phi_i > 0 \cap l_{i1} > l_{i2}) \cup (\phi_i < 0 \cap l_{i1} < l_{i3}) \end{cases} \quad (5)$$

Subsequently, the pose vector of the end-actuator with respect to the

local coordinate  $\mathbf{p}_i \in \mathbb{R}^3$  is figured out

$$\begin{aligned} \mathbf{p}_i &= [p_i^x, p_i^y, p_i^z]^T \\ &= [c\phi_i(1 - c\theta_i)/\kappa_i, s\phi_i(1 - c\theta_i)/\kappa_i, s\theta_i/\kappa_i]^T \end{aligned} \quad (6)$$

where  $c \triangleq \cos$  and  $s \triangleq \sin$ .  $\theta_i$  is the central angle of the backbone, which can be easily calculated by  $\theta_i = L_i/\kappa_i$ . The pose of the end-actuator is simulated in Fig. 5 to show the working space for the unit-segment manipulator.

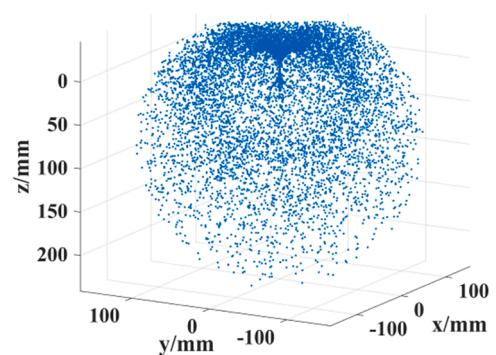
The improved D-H method is employed to calculate the homogeneous transformation matrix (HTM) to construct the spatial attitude transformation between adjacent segments (see Fig. 6).

Through a series of orderly rotation and translation operations, the HTM  $\mathbf{T}_{i+1}^i \in SE(3)$  from segment  $i$  to segment  $i+1$  is derived in (7). More details are in [12].

$$\begin{aligned} \mathbf{T}_{i+1}^i &= \mathbf{R}^z(\phi_i)\mathbf{P}^x(1/\kappa_i)\mathbf{R}^y(\theta_i)\mathbf{P}^x(-1/\kappa_i)\mathbf{R}^z(-\phi_i) \\ &= \begin{bmatrix} \mathbf{R}_{i+1}^i & \mathbf{p}_i \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \end{aligned} \quad (7)$$

where  $\mathbf{R}^z, \mathbf{R}^y \in \mathbb{R}^{4 \times 4}$  are the homogeneous rotation matrices of the  $Z_i$  and  $Y_i$  axes, respectively.  $\mathbf{P}^x \in \mathbb{R}^{4 \times 4}$  is the homogeneous translation matrix along the  $X_i$  axis. In addition,  $\mathbf{R}_{i+1}^i \in SO(3)$  and  $\mathbf{p}_i$  are the orientation matrix and pose vector of the end-actuator along the backbone. The expression of the matrix  $\mathbf{R}_{i+1}^i$  is described as

$$\mathbf{R}_{i+1}^i = \begin{bmatrix} c^2\phi_i(c\theta_i - 1) + 1 & s\phi_i c\phi_i(c\theta_i - 1) & c\phi_i s\theta_i \\ s\phi_i c\phi_i(c\theta_i - 1) & s^2\phi_i(c\theta_i - 1) + 1 & s\phi_i s\theta_i \\ -c\phi_i s\theta_i & -s\phi_i s\theta_i & c\theta_i \end{bmatrix} \quad (8)$$



**Fig. 5.** The working space for the unit-segment manipulator.

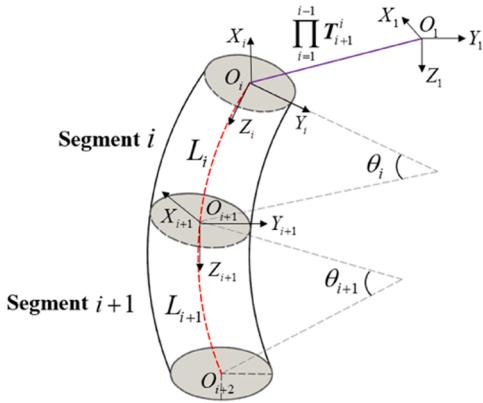


Fig. 6. The multi-section model.

Multiply  $\mathbf{R}_{i+1}^i$  for each segment sequentially, the orientation matrix  $\mathbf{R}_i$  relative to the global coordinate  $\{O_1-X_1Y_1Z_1\}$  is yielded as

$$\mathbf{R}_i = \begin{cases} \mathbf{R}_{i+1}^i, i = 1 \\ \mathbf{R}_{i-1}\mathbf{R}_{i+1}^i, i > 1 \end{cases} \quad (9)$$

Furthermore, the recursive formula of the pose of the end-actuator in the global coordinate is obtained accordingly.

$$\mathbf{P}_i = \begin{cases} \mathbf{p}_i, i = 1 \\ \mathbf{P}_{i-1} + \mathbf{R}_{i-1}\mathbf{p}_i, i > 1 \end{cases} \quad (10)$$

In a word, the kinematics for the multi-segment soft manipulator is generated iteratively to accomplish the entire mapping from joint space to the workspace.

### 3.3. Advantages and limitations

The CC kinematics is the most classical and widely used model to estimate the 3D shape of the soft manipulator. Some significant advantages are summarized here. (1) Simple and efficient: Without considering the coupling between PMAs and the influence of any force on kinematics, the model becomes simple. Also, inverse kinematics can be easily obtained by the inverse process of the above method. There are only three control variables  $\mathbf{q}_1$  in the control system, which is conducive to the controller design. It has the low-cost computation of about 9 ms, which is beneficial to real-time control; (2) Universal and robust: This method is suitable for many kinds of soft manipulators, as long as the same sensor distribution is observed. More importantly, it can still complete effective shape estimation in anywhere, such as seabed and narrow gaps [33], instead of being constrained by environment.

As for limitations, CC kinematics introduces singularities to the numerical calculation of zero curvature section, so it is impossible to simulate pure elongation pose. When all elements in  $\mathbf{q}_1$  are equal, the numerator and denominator of Eq. (4) are both 0, rendering all

kinematics model containing  $\phi_i$  undefined [34]. Fig. 7 shows the numerical solution for  $\mathbf{p}_i$  when  $l_3 \rightarrow l_1 = l_2 = 100\text{mm}$ .  $p_i^x$  and  $p_i^y$ , mentioned by (6), produce unreliable results across the singularity neighborhood, because their formulas contain  $\phi_i$  while  $p_i^z$  does not. But their numerical solutions both have a number of invalid values near the singularity. Godage [8] proposed the Taylor expansion approximation method to avoid singularities. Although it has strong numerical stability, it needs 11 order Taylor expansion for each element of HTM, and the computation is intensive.

Moreover, since CC kinematics only considers the distance constraint between PMAs and ignores the fact that they are also strictly mechanically coupled in position and orientation, as well as the overly high aspect ratio design for prototypes, almost all prototypes deviate from the CC deformation and twists into complex 3D curves, especially in the case of large deformation. Some experimental studies based on RSES are completed to explain this limitation further. We select three representative poses as given in Fig. 8.

Due to the slight deformation of pose 1, the prototype is still in the shape of constant curvature, and the yellow curve almost coincides with the red curve. The relative simulation result computed by RSES is strongly matched with the prototype in Fig. 8a. When the deformation of the prototype continues to increase, the red curve deviates from the yellow curve in pose 2, and the simulation results can approximate the shape of the prototype (see Fig. 8b). It follows that the prototype has formed slight 3D torsion at this time, which RSES cannot express. As seen in Fig. 8c, the prototype with great deformation produces a large 3D torsion, resulting in the yellow curve far away from the red curve. The shape difference of the simulation is obviously visible for pose 3, so the RSES results are no longer of the reference value.

The RSES results for the multi-segment manipulator are presented in our previous work [35]. We believe that the error caused by the constant curvature model accumulates as the number of segments increases. In brief, there is a clear need for a better method to recognize the shape of the soft manipulator.

## 4. ASES based on vision

In this section, ASES based on stereo vision is raised without any mathematical model. Contour extraction, centerline clustering, and 3D reconstruction are introduced. Finally, some experiments and simulations are also given to analyze the advantages and limitations of this method.

### 4.1. Contour extraction: image preprocessing

The purpose of image preprocessing is to segment the manipulator data from the image data we captured. The single white is selected as the experimental background to easily locate the manipulator. The specific process is as follows.

Firstly, the camera calibration is performed by the factory parameters provided in ZED-API, and distortion-free images with a resolution of  $1280 \times 720$  are acquired. Secondly, a Gaussian filter is used to suppress the noise of the acquired images. We convert the denoised images to HSV and then perform mask processing to get the binary images, where prototype's color is set as a threshold. In addition, morphological filter is used to remove isolated pixels, resulting in a smoother edge. Finally, we wish to extract the optimal encompassing contour about the manipulator for real-time design. In this work, the Canny edge detection algorithm, supported by OpenCV, is provided for contour extraction to get a set of pixels points almost completely along the edge of the manipulator. The image preprocessing result is shown in Fig. 9 (take the left image as an example).

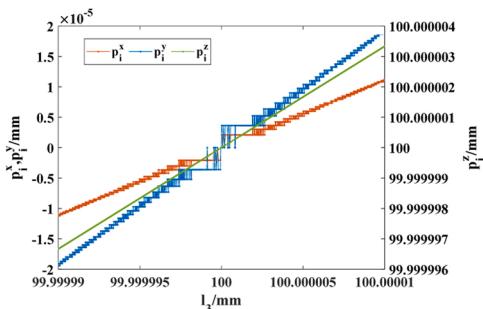
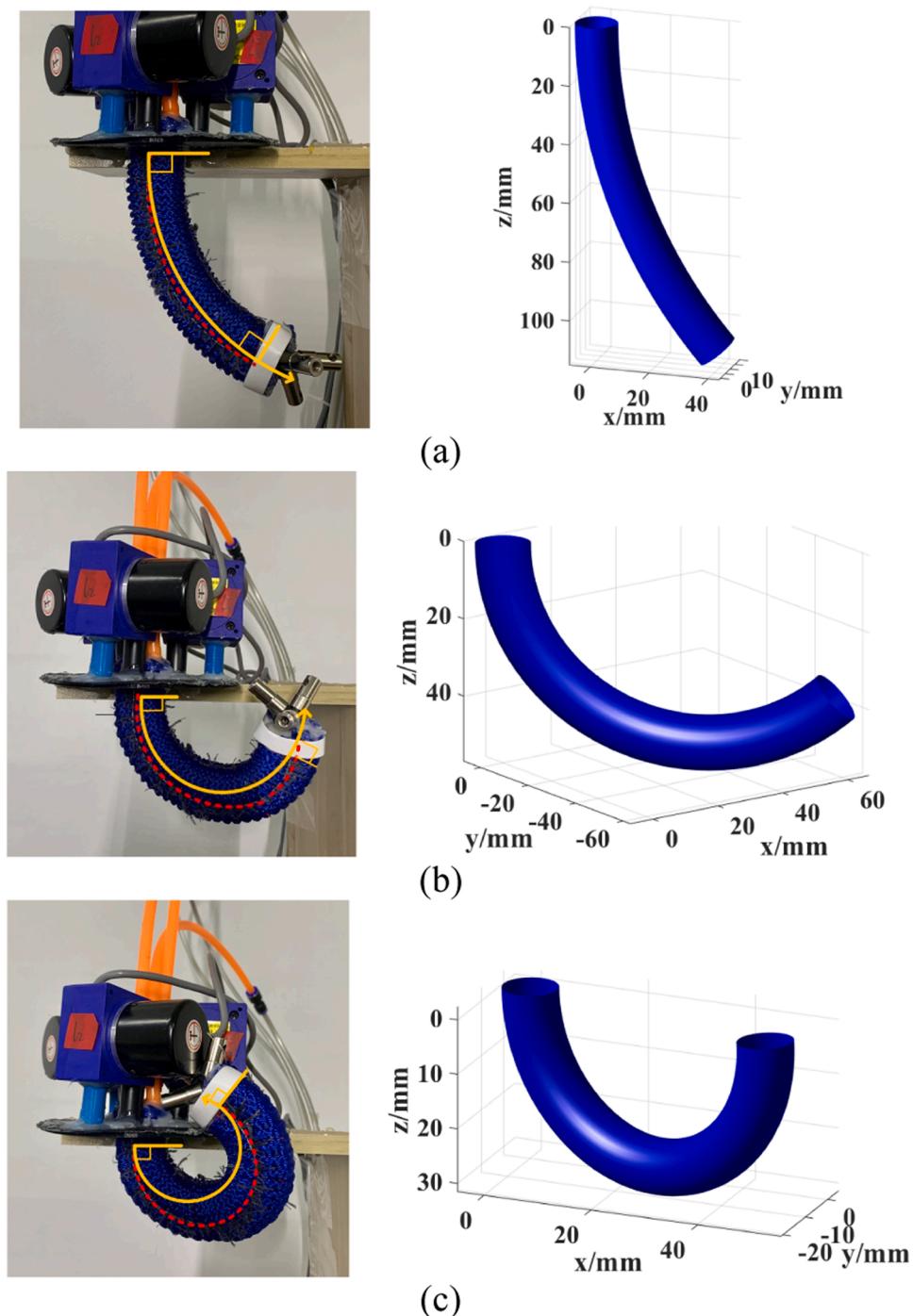


Fig. 7. The numerical solution of  $\mathbf{p}_i$  for  $l_3 \pm 10^{-5}$ .



**Fig. 8.** The RSES results for three poses compared to the experimental results. The yellow curve printed on the image is the 2D CC shape of the current attitude, and the red one is the real 2D shape depicted along the gap between PMAs. For a more realistic display, the centerline is substituted by the tube to simulate the 3D shape of the soft manipulator. (a) Pose 1. (b) Pose 2. (c) Pose 3.

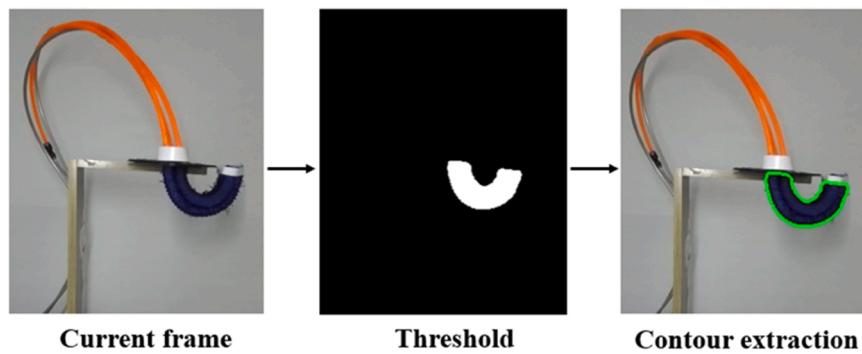
#### 4.2. Clustering: SOM algorithm

Because the soft manipulator is a uniform and symmetrical tubular object, the centerline as the backbone can well reflect its spatial shape. SOM algorithm is adopted to identify the centerline from the contour of the manipulator in the image plane. SOM neural network is composed of the input layer and output layer only, as shown in Fig. 10.

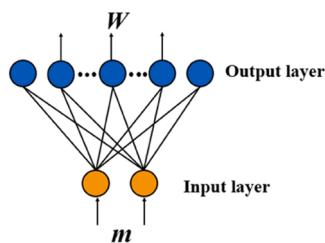
The number of input neurons  $a$  corresponds to the dimension of the input sample. Thanks to the input of SOM is the contour data of the manipulator, here,  $a = 2$ . Where  $h$  is the number of contour data and.. is the 2D coordinates of each contour data. As for output neurons, it

represents the center points.. formed by clustering, where  $b$  is the expected number of center points and  $\mathbf{W}_k \in \mathbb{R}^2$  means the 2D coordinates of each center point. The training process of the SOM algorithm can be summarized into the following steps.

- (1) **Initialize parameters:** Initialize  $\mathbf{W}$ , and set the neighborhood radius  $S$ , initial learning rate  $\alpha(0)$  and the training number  $\lambda$ .
- (2) **Find the winning neurons:** Randomly select an input  $\mathbf{m}_r$  and calculate its similarity with each output  $\mathbf{W}_k$ , expressed by Euclidean distance. The output neuron with the smallest



**Fig. 9.** Image preprocessing results



**Fig. 10.** SOM neural network structure.

Euclidean distance is recorded as the winner neuron  $k^*$ . The equation is defined:

$$k^* = \arg \min_{k \in [1,b]} (\|\mathbf{m}_r - \mathbf{W}_k\|) \\ = \arg \min_{k \in [1,b]} \left( \sqrt{(\mathbf{m}_r - \mathbf{W}_k)^T (\mathbf{m}_r - \mathbf{W}_k)} \right) \quad (11)$$

- (3) **Design winning neighborhood:** the winning neighborhood  $S_{k^*}(t)$ , with  $k^*$  as the center and  $S$  as the radius, decreases gradually with time  $t$ .
  - (4) **Update weight:** The output neuron updates the weight based on the gradient descent method in  $S_{k^*}(t)$ . The update rate is satisfied:

$$\mathbf{W}_k(t+1) = \mathbf{W}_k(t) + \eta(\delta)\alpha(t)[\mathbf{m}_r - \mathbf{W}_k(t)], k \in S_{k^*}(t) \quad (12)$$

where the neighborhood function  $\eta(\delta)$  meets that the farther the topological distance  $\delta$  from.. is, the smaller it will be, usually

represented by Gaussian function.  $\alpha(t)$  often uses  $\alpha(0)/(1+t/\lambda)$  to indicate that it decreases monotonically with  $t$ .

- (5) **Judge termination:** Check whether the current training number reaches  $\lambda$ . If not, increase the training number by 1, return to step (2), continue training until the  $\lambda$  is reached, and output  $\mathbf{W}_k$ .

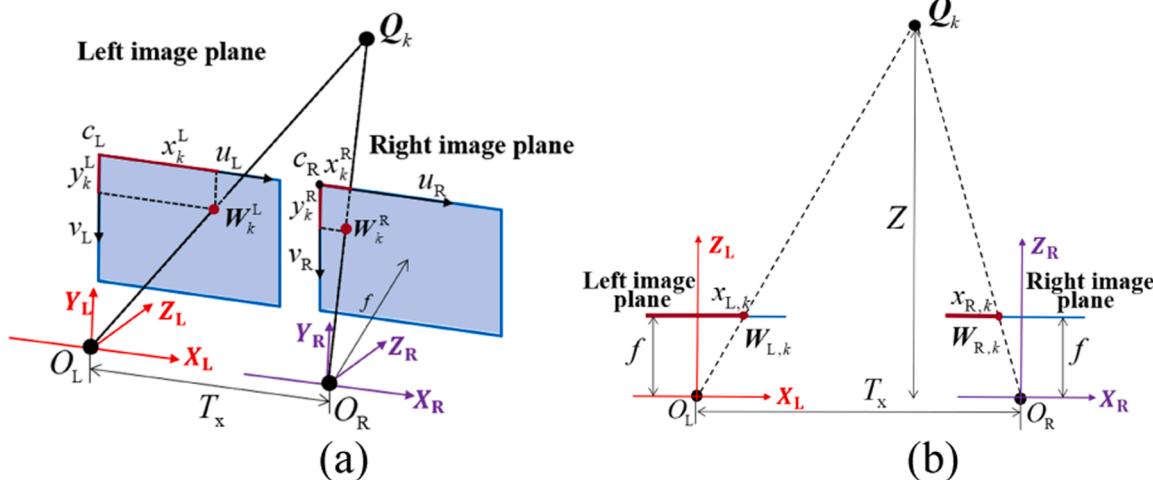
On account of the ordering guarantee property of SOM algorithm,  $\mathbf{W}_k^L$  and  $\mathbf{W}_k^R$ , from the left and right images separately, can be automatically matched to correspondence point without any sorting algorithm.

### 4.3. 3D construction: triangulation

It is necessary to establish a projection relationship between the 3D space where the manipulator resides and the image plane we have acquired. In view of the special structure that left and right cameras of ZED are strictly parallel, the triangulation method is employed to calculate the disparity of the point on the left and right image plane through the geometric model, so as to obtain its depth, as shown in Fig. 11. The distance between the optical centers of the left  $O_L$  and right cameras  $O_R$  is the baseline length  $T_x$ , and the focal length  $f$  satisfies  $f = f_L = f_R$ . The left  $c_L - u_L v_L$  and right pixel coordinate system  $c_R - u_R v_R$  is set up as follows.

$\mathbf{W}_k^L$  and  $\mathbf{W}_k^R$  are the projections of a backbone  $\mathbf{Q}_k = [Q_k^x, Q_k^y, Q_k^z]^T \in \mathbb{R}^3$  on the left and right image planes respectively, and their x coordinates are noted as  $x_k^L$  and  $x_k^R$ . According to the geometric principle of spatial triangle similarity, the depth  $Q_k^z$  is solved:

$$Q_k^z = fT_x/\tau_k \quad (13)$$



**Fig. 11.** Triangulation model. (a) Imaging schematic diagram. (b) Geometric diagram.

where  $\tau_k = x_k^L - x_k^R$  is the disparity. On the basis of the depth model, the dimension information of the X and Y axes can be calculated by the same way

$$Q_k^x = x_k^L Q_k^z / f \quad (14)$$

$$Q_k^y = y_k^L Q_k^z / f \quad (15)$$

The backbone  $\mathbf{Q} = [\mathbf{Q}_1, \dots, \mathbf{Q}_b] \in \mathbb{R}^{3b}$  is generated iteratively in turn. The left camera coordinate  $\{O_L - X_L Y_L Z_L\}$  is the default coordinate here. For subsequent research, the coordinate definition needs to be consistent with the kinematics method (see Section 3.2). The coordinate transformation is completed below

$$\begin{bmatrix} \hat{\mathbf{Q}} \\ \mathbf{1} \end{bmatrix} = \mathbf{T}_{L-g} \begin{bmatrix} \mathbf{Q} \\ \mathbf{1} \end{bmatrix} \quad (16)$$

where  $\mathbf{T}_{L-g} \in SE(3)$  is HTM from the left camera coordinate  $\{O_L - X_L Y_L Z_L\}$  to the global coordinate  $\{O_1 - X_1 Y_1 Z_1\}$ , determined by Ordinary Least Squares with multiple known markers. Ultimately, the set of  $\hat{\mathbf{Q}} = [\hat{\mathbf{Q}}_1, \dots, \hat{\mathbf{Q}}_b] \in \mathbb{R}^{3b}$  is the final result of ASES.

#### 4.4. Advantages and limitations

Compared with the other shape estimation method that requires modeling, machine vision is a low-cost, high-precision and non-contact measurement. To analyze the performance of the ASES, various experiments were performed on the soft manipulator platform. The pressure of the three PMAs is adjusted to actuate the prototype into the variable curvature poses. As shown in Fig. 12a, we selected four typical poses and ran ASES to sense their 3D shape. The results of contour extraction and SOM are shown in Fig. 12b. The parameters involved in the SOM algorithm are set in the Table 1 to ensure well clustering results.

The final 3D reconstruction results are simulated in Fig. 12c, and each pose is distinguished by different symbols. The improved cubic spline fitting is applied to better express the 3D shape in the form of a continuum through each data point. It is observed that the simulation results of ASES agree very well with the experimental results, which provides a great solution to inaccurate shape estimation under variable curvature poses.

Remark The measurement error from ZED is not considered.

The limitations of vision are apparent. It has strict environmental requirements, such as lighting quality and background complexity, which directly affect the effectiveness of the detection. Therefore, the vision method is only applicable to the indoor experimental environment where the conditions can be arranged artificially. As seen in Fig. 12a, the light is supplemented throughout the above experiment to guarantee the correct contour extraction. For comparison, the same experiments were conducted under poor light. As given in Fig. 13a, the bad contour leads to the subsequent center point clustering being no longer effective.

Not only that, vision method will also lead to failure when the prototype appears self-occlusion, which greatly limits the application. In Fig. 13b, the prototype is bent backward so that the contour cannot be correctly identified. At this time, ASES should not be utilized.

### 5. DNN approach and experimental verification

In this section, a deep learning-based approach is implemented to explore the highly non-linear complex relationships between RSES and ASES. Several experiments have been done to evaluate the performance.

#### 5.1. DNN based approach

As stated above, RSES and ASES have their own advantages. Assuming that the results of ASES can be obtained by only using RSES,

the advantages of the two estimation methods are perfectly combined. In other words, it comes down to finding the highly non-linear mapping between  $\mathbf{q}_1 \in \mathbb{R}^3$  and  $\hat{\mathbf{Q}} \in \mathbb{R}^{3 \times 7}$  (take unit-section manipulator as an example). However, most of the non-linear mappings are hard to describe with mathematical formulas. To settle this circumstance, DNN, as a general approximator, is a practical modeling technique to provide an approximate solution for  $g(\cdot)$  by training a large number of input/output data. It can be written concisely a

$$\tilde{\mathbf{Q}} = g(\mathbf{q}_1) \quad (17)$$

where  $\tilde{\mathbf{Q}} \in \mathbb{R}^{3 \times 7}$  is infinitely close to  $\hat{\mathbf{Q}}$  in theory. The architecture of DNN adopted in this work is displayed in Fig. 14. It takes  $\{l_{11}, l_{12}, l_{13}\}$  (each element of  $\mathbf{q}_1$ ) as input neuron and serves each element of  $\hat{\mathbf{Q}}$  as the output neuron. Multiple hidden layers are composed of full connection layers, and each node possesses an activation function to add nonlinearity. The forward-propagation algorithm can be summarized as

$$\mathbf{C}_n = \varphi(\mathbf{V}_n \mathbf{U}_{n-1} + \mathbf{B}_n) \quad (18)$$

where  $\mathbf{B}_n$  is expressed as the bias vector for layer  $n$ ,  $\mathbf{V}_n$  is the weighting matrix connecting layer  $n-1$  to layer  $n$ , then the network parameter is set as  $\beta = \{\mathbf{V}, \mathbf{B}\}$ .  $\mathbf{U}_{n-1}$  and  $\mathbf{C}_n$  are the output vector for layer  $n-1$  and layer  $n$  separately. Additionally,  $\varphi(\cdot)$  represents the activation function.

The entire learning process of DNN can be divided into three main steps:

#### (1) Dataset collection:

DNN is regarded as a data-driven modeling method, so the enormous dataset allows the trained model to have better accuracy. In this work, the dataset is generated on our experiment platform. We have provided varying pressure for three PMAs to explore the real workspace of the soft manipulator. The DAC is programmed to control  $\mathbf{P}^C \in \mathbb{R}^3$  to change discretely from 0 bar to 3 bar with a fixed step of 0.1 bar. For each actuation,  $\hat{\mathbf{Q}}$  is figured out to save as an output sample by running ASES. Meanwhile, The RSES measures  $\mathbf{q}_1$  by cable encoders as an input sample. In this manner, a group of input/output samples is memorized. The above sampling process lasted approximately 35 h and resulted in a total of  $N_{total} = (31)^3 = 29791$  samples. Some samples need to be removed (about 827) when a self-occlusion occurs on the manipulator and causes the ASES to fail. For the integrity of the dataset, the artificial samples can be created by rotating around the z-axis to supplement removed samples on account of the radial symmetrical distribution of PMAs. The final dataset is randomly distributed into three parts: training set (70%), test set (15%) and validation set (15%).

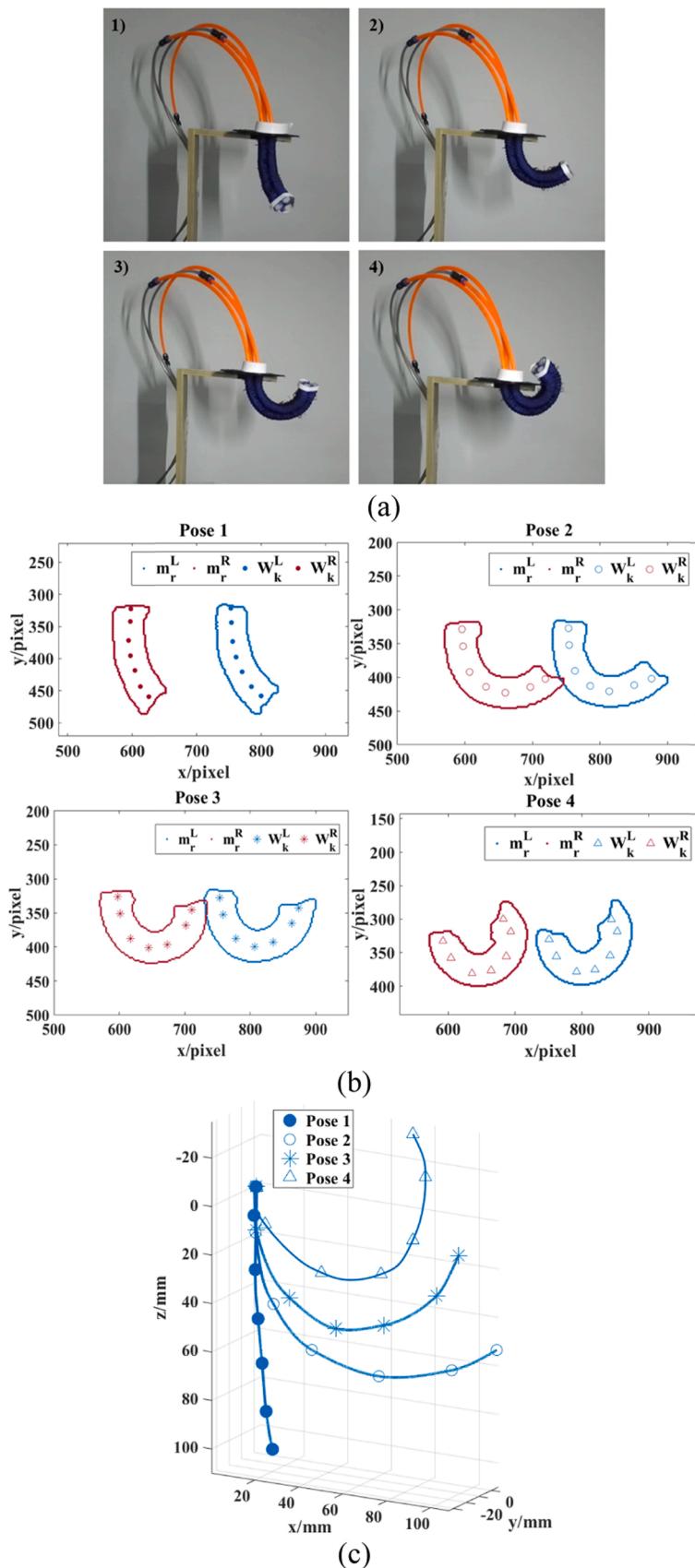
#### (2) Network training:

The purpose of network training is to search the ideal network parameters  $\beta^*$  to minimize the Mean Square Error (MSE) between  $\tilde{\mathbf{Q}}$  and  $\hat{\mathbf{Q}}$  where is from the dataset. It can be concluded as:

$$\begin{aligned} \beta^* &= \arg \min_{\beta} G(\hat{\mathbf{Q}}, \tilde{\mathbf{Q}}) \\ &= \arg \min_{\beta} \frac{1}{N_{tr} N_{out}} \left( \sum_{i=1}^{N_{tr}} \left\| \hat{\mathbf{Q}}^{(i)} - g_{\beta}(\mathbf{q}_1^{(i)}) \right\|_2^2 + \gamma \|\beta\|_2^2 \right) \end{aligned} \quad (19)$$

where  $\|\cdot\|_2$  is noted as L2 norm.,  $N_{tr}$  is the number of the output neurons, and  $N_{tr} = 20854$  is the number of training samples.  $G(\cdot)$  implies the MSE loss function we choose. It is worth noting that L2 regularization is added to the loss function as a penalty term and decay the magnitude of  $\beta$  by means of adjusting  $\gamma$ , which can effectively reduce the complexity of the DNN model and prevent overfitting. Here,  $\gamma = 0.01$ .

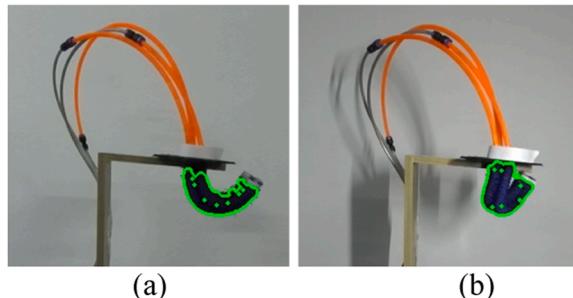
Referring to the MSE, the Adam optimizer with momentum term is operated to dynamically update, through backpropagation until MSE converges to the minimum value. In each training epoch, the training set



**Fig. 12.** The ASES results for four typical poses. (a) Experiment results with four typical poses of the soft manipulator. (b) Simulation results with contour extraction and SOM. (c) Simulation results with 3D shape estimation.

**Table 1**  
SOM algorithm parameter setting.

$S$	$\alpha(0)$	$\lambda$	$b$
3	0.01	15	7



**Fig. 13.** Experimental results in some bad cases. The contours and center points marked in green are printed to the image. (a) poor light conditions. (b) self-occlusion.

follows the above training process in batches, while the validation set is only performed the forward propagation with fixed  $\beta$ , and takes MSE as the criterion to judge whether the current network structure and hyperparameter are reasonable, which improves the generalization of the model. The topological structure and hyperparameter configuration of final DNN we chosen is given in Table 2 (maybe not the best, but enough).

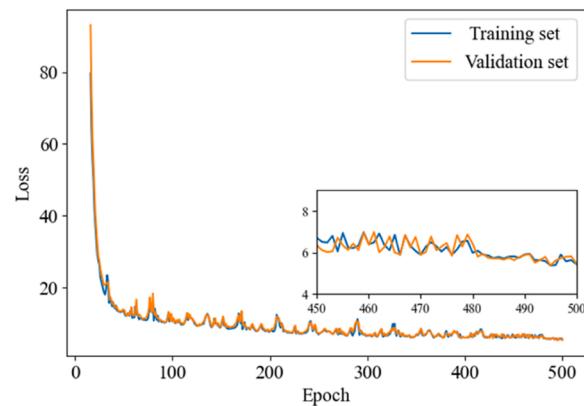
Fig. 15 shows the performance of the loss function on the training set and the validation set for each training epoch. Obviously, the loss function dropped significantly within the first 20 epochs of training, which explains that  $\beta$  at this time are rapidly moving to  $\beta^*$  by a large margin. When the training reaches about 500 epochs, the loss function has converged steadily and has no further downward trend. Therefore, the training would cease, and the DNN model is saved.

(3) **Performance evaluation:** The performance of the trained DNN on the test set is further evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as the new error criteria. MAE can more directly reflect the real error between  $\tilde{Q}$  and  $Q$ , and RMSE is more sensitive to the dispersion of samples. As listed in Table 3, we developed the error analysis for the test set in the dimensions of x, y and z (all samples in mm). In the two error criteria, the error on the z-axis is slightly larger than that of the other two axes.

Nevertheless, sometimes the overall error evaluation of data cannot investigate the stability and universality of the DNN for any case thoroughly. As a consequence, we made an in-depth error analysis for each sample. As shown in Fig. 16, the three frequency histograms for the error

**Table 2**  
The hyperparameter configuration of final DNN.

Layer name	Topological structure		Hyperparameter		
	Composition	Layer name	Composition	Item	Setting
Input layer 1	Input [3,15]	Hidden layer 5	Linear layer [15,45] +RELU	Learning rate	0.001
Hidden layer 2	Linear layer [15] +RELU	Hidden layer 6	Linear layer [45] +RELU	Optimizer	Adam
Hidden layer 3	Linear layer [15] +RELU	Hidden layer 7	Linear layer [45] +RELU	Epoch	500
Hidden layer 4	Linear layer [15] +RELU	Output layer 8	Output layer [21,45]	Batch size	128

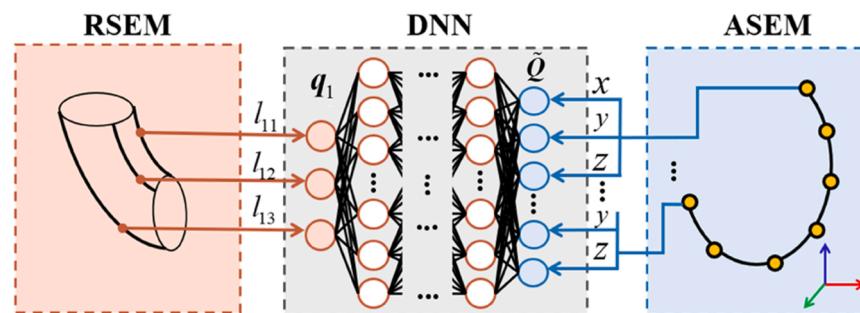


**Fig. 15.** The performance of the loss function on the training set and the validation set.

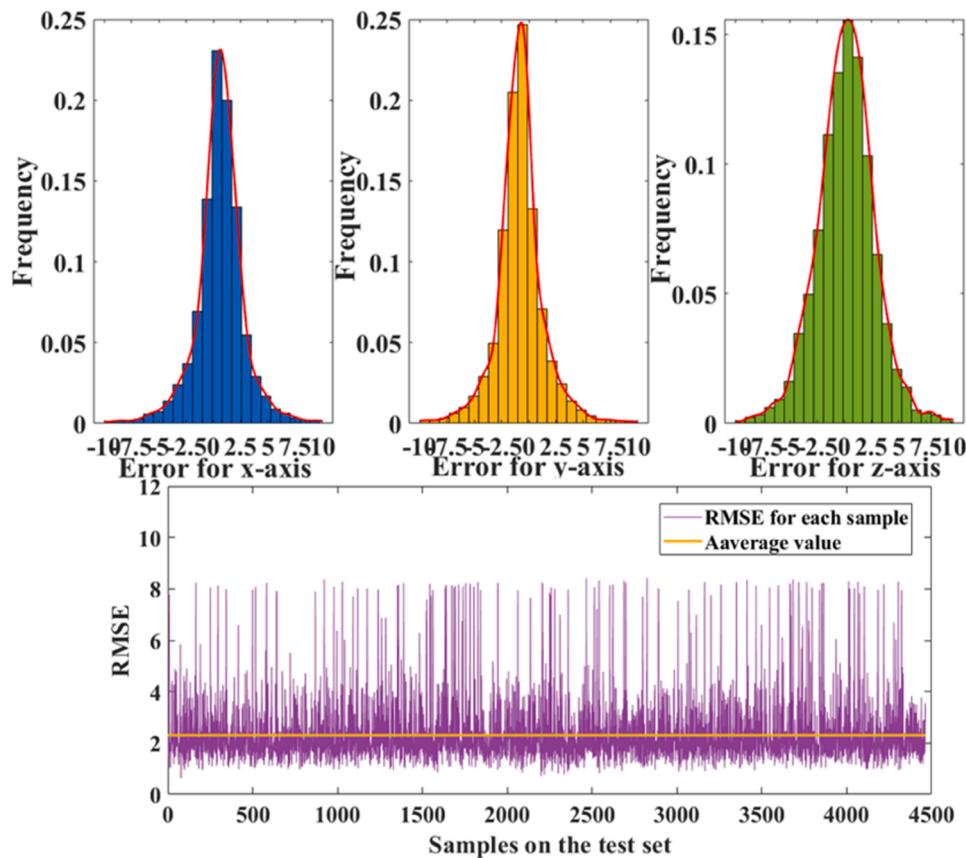
on the test set deliver the error distribution for each sample on the x, y and z axes. It can be observed that the error on the three axes does not exceed  $\pm 10$  mm, and the proportion of samples with error within  $\pm 2.5$  mm on the x, y, and z axes reach 82%, 81.4% and 73.75% of the total test set, individually. The number of samples with an error of about  $\pm 10$  mm only accounts for 0.11%, 0.18% and 0.15% of the test set. It

**Table 3**  
Error evaluation of trained DNN on the test set.

Axis	MAE	RMSE
x-axis	1.7601	2.5563
y-axis	1.8049	2.5850
z-axis	2.0677	2.6141



**Fig. 14.** The principle of the DNN approach.



**Fig. 16.** The detailed error distribution on the test set. The red lines in the three histograms are computed by spline interpolation with the frequency of each error interval.

implies that although the performance on the z-axis is slightly inferior to the other two, the error distributions of the three dimensions are both similar to the Gaussian distribution, which is what we want to get. More concretely, we carried out the RMSE of each sample on the test set. Among the 4463 samples, the RMSE of only 60 samples is approximately 8. As described by the yellow line in Fig. 16, the average RMSE for all samples is 2.302.

To sum up, the trained DNN model has strong robustness for most samples on the test set and reflects satisfactory generalization performance. In Fig. 17, the samples with the minimum, average and maximum RMSE and the corresponding learning results are given to visualize the performance of this learning model for 3D shape estimation. In the view of the displayed performance, the learning result with maximum RMSE on the test set still falls in an acceptable range.

## 5.2. Experimental verification

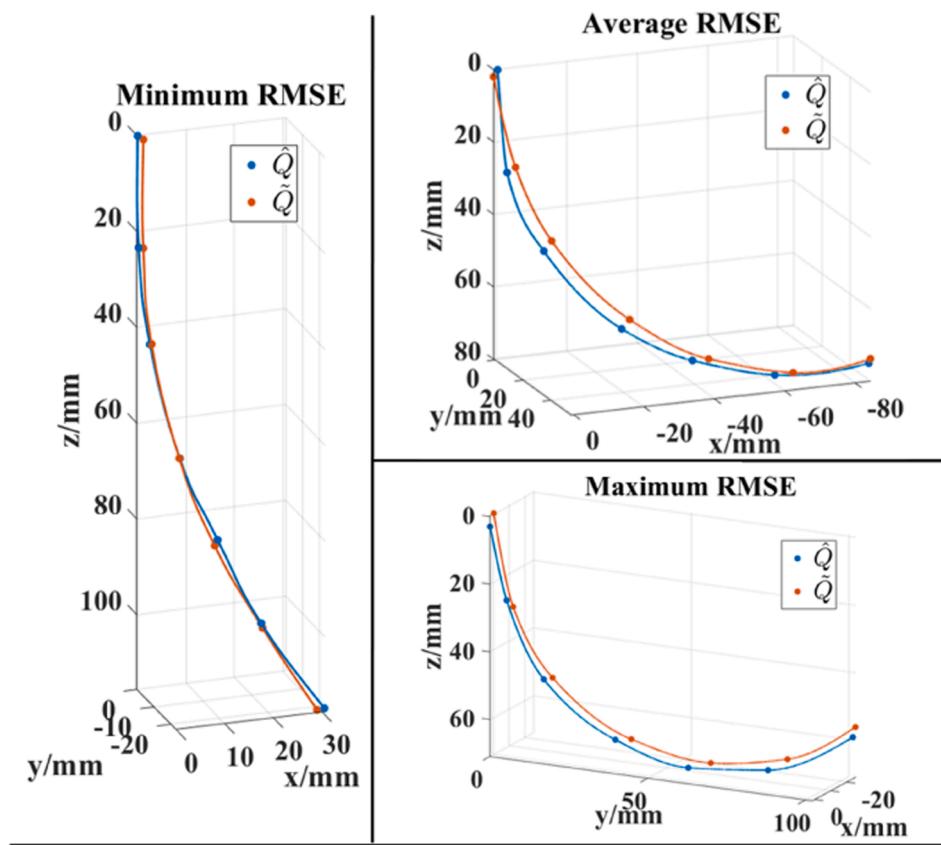
In addition to the verification on the test set, a real-time bending deformation experiment was also conducted to prove the dynamic performance of DNN. The continuous bending was generated by pressure variation for PMA2 from 1 bar to 2.5 bar with a fixed step of 0.05 bar, and RSES and ASES were operated simultaneously during deformation. In order to coordinate the synchronization of data obtained by the two detection systems, we reduced the sampling frequency of STM32 for  $q_1$  by the delay function to match the frame of the camera. The results of contour extraction with equal time intervals during motion by ASES are shown in Fig. 18a. The red contours indicate the starting pose and the final pose. The SOM results with the corresponding pose are shown in Fig. 18b. We also give the results of ASES and DNN for the 3D shape estimation of the soft manipulator (see Fig. 18c). The prediction effect of DNN seems to be great among these poses.

For simplicity, we only exhibited the trajectory of the end-actuator instead of the entire shape to analyze the dynamic process. The whole motion process lasted for 30 s, and the three methods all produced 105 trajectory points, as shown in Fig. 18d. As you can see,  $\tilde{Q}_7$  are able to continuously approach to the  $\hat{Q}_7$  in a high accuracy, while the results of RSES have a large deviation in the latter half of the motion process. Some 3D shapes estimated by RSES during motion are also plotted here. The change in  $q_1$  is recorded in Fig. 18e.

Next, error analysis is performed to quantify the accuracy of the two methods (take the results of ASES as the standard value). In Fig. 19, the error of DNN in each dimension has been maintained at a low level throughout the movement, but the error of RSES is satisfactory only in the first 5 s. It can be found that the RSES error on the y-axis begins to increase after about 15 s, because the prototype deforms greatly and begins to produce 3D torsion, consistent with the deviation of the RSES trajectory in Fig. 18d.

The RMSE for the two methods in each dimension is listed in Table 4. The DNN solution has achieved good performance, which confirms its effectiveness and accuracy even in real-time experiments. On the contrary, RSES is less accurate, resulting from the inherent constant curvature modeling assumption. Specifically, the constant curvature model considers the three cables to be coplanar with the centerline and cannot capture some indescribable properties, for example, hysteresis, viscoelasticity and friction between cable and nylon mesh. The prototype is made of non-linear flexible materials without the characteristics of CC deformation, which also inherits the memory effect and nonstationary behavior caused by rubber, and the torsional effect is also ignored in the modeling process. Therefore, RSES cannot meet the needs of high-precision shape estimation in most cases.

Some attempts have also been made to explore different kinds of neural networks in solving this work. Two common neural networks,



**Fig. 17.** The DNN results and the corresponding vision results with the minimum, average and maximum RMSE on the test set.

Radial Basis Function (RBF) and Multilayer Perceptron (MLP) neural network with two hidden layers, are selected. Use the same dataset and epochs to train RBF and MLP neural network to verify if they can also fit the complex nonlinear relationship between RSES and ASES equally well. The training process is shown in Fig. 20.

There is a convergence trend in the overall training process of the MLP neural network, but obviously, 500 epochs are not enough for it, and the loss error is relatively large compared to DNN. This is because the number of hidden neurons in MLP neural networks is small so that the ability to fit complex functions is limited. However, the loss error of the RBF neural network hardly sees a significant decrease, indicating that the fitting of this neural network fails, and is not suitable for this work. In summary, DNN is more suitable for the research work in this paper. We are interested in finding more potential neural network structures to further improve the accuracy of our work.

### 5.3. Discussions

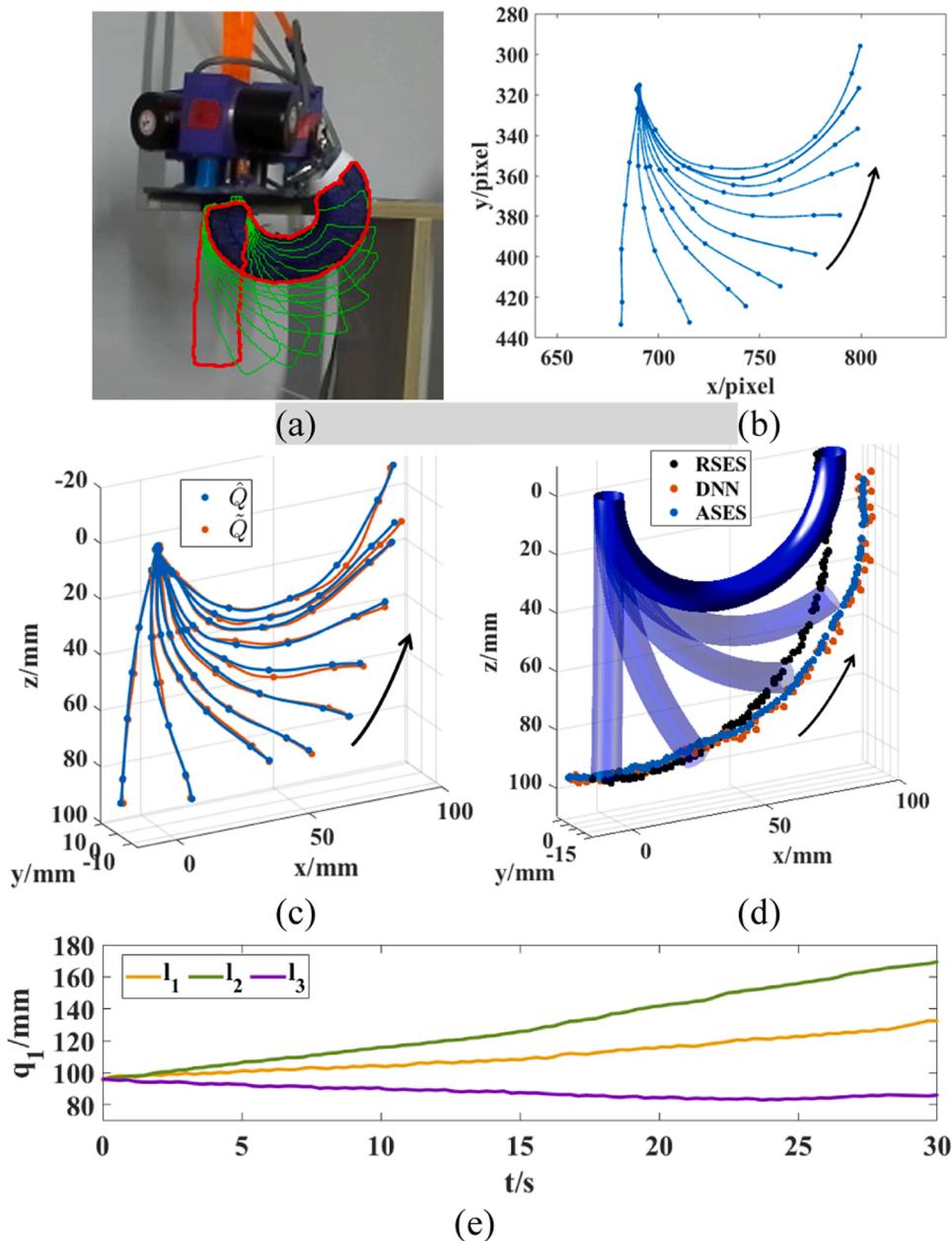
It can be concluded that the proposed learning-based approach can obtain accurate visual results only by relying on the simplest RSES, which is undoubtedly a better choice among shape estimation frameworks. Most often, there are inevitably some limitations. (1) To ensure the accuracy of DNN, it is necessary to establish a large number of datasets to cover all possible situations, which means that tens of thousands of offline experiments are carried out at high frequency. The process of data collection has a high time cost and is also regarded as a test for the durability of the prototype. If extended to a multi-segment manipulator, the size of the dataset grows exponentially with the amount of control variable, which becomes difficult to build by experiment. Furthermore, the data collected by the experimental method may be unreliable or biased due to the nonlinearity of the soft manipulator. (2) DNN contains many adjustable hyperparameters that are artificially

set, which is a long phase to find a suitable set of hyperparameters, because each change of hyperparameters requires restarting the training process, while DNN can only guarantee good convergence in the local optimal area, rather than the global optimum. (3) This work establishes a shape estimation model for the prototype in a free state during the experiment, but in practical applications for the restricted environment, it often interacts with the environment or requires additional external loads, resulting in a decrease in the accuracy of the trained model. Even in the free state, flexible materials generally deform slightly after long-term use, which leads to a slow decrease in the accuracy of the trained DNN model. If the same new prototype is replaced, the database will also be re-established and re-trained. This is because the prototypes produced by the manual process have certain differences in each production.

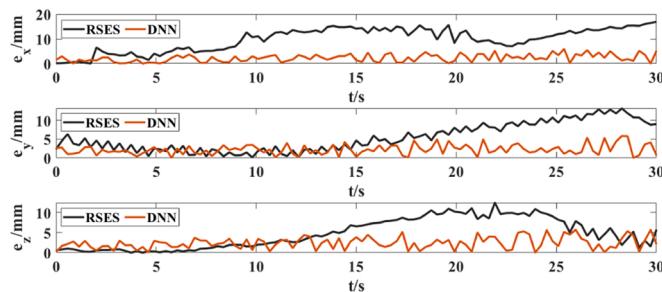
In the future work, we intend to use the hybrid learning method to establish a simulation dataset through the finite element method, so as to free the prototype from the training process and facilitate the shape estimation of the multi-segment soft manipulator. And more complex and realistic environments will be introduced for experimental evaluation.

### 6. Conclusion

In this work, a DNN approach is proposed to accurately estimate the 3D shape of the soft manipulator in real time, which integrates traditional sensing, machine vision, and deep learning. The CC kinematics-based RSES has the singularity and the limitation of CC, which can only achieve the effect of rough shape estimation. ASES based on vision can accurately estimate the 3D shape of the soft manipulator, whereas the extremely strict requirements for the environment and self-occlusion limit its application. Thus, a novel learning-based framework using DNN is adopted to fitting the non-linear mapping between ASES and RSES.



**Fig. 18.** The results of bending deformation experiments. (a) Contour extraction. (b) Centerline clustering. (c) The results of ASEM and DNN. (d) Trajectory generated by three methods. (e) The change trend of  $q_1$ .



**Fig. 19.** Absolute error curves of two methods in x, y, z axes.

The error evaluation for the trained DNN is conducted on the test set. The results show that the RMSE of the x, y and z axes are within 2.7, and the error distribution is similar to the Gaussian distribution. A real-time

**Table 4**  
RMSE for two methods in each dimension.

Axis	RSES	DNN
x	10.8861	2.2124
y	6.5395	2.1071
z	6.5560	2.2998

bending deformation experiment is also performed to suggest the DNN approach can accurately measure 3D shape for soft manipulator via length information in real time, and the RMSE is always maintained within 3.

### **Declaration of Competing Interest**

The authors declare that they have no known competing financial

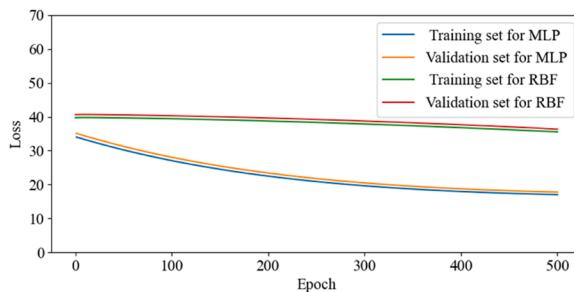


Fig. 20. Training process with the RBF and MLP neural networks.

interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by the Open Fund for Space Intelligent Control Technology Laboratory (HTKJ2020KL502014) and the National Key R & D Plan (2020YFB1506702).

## References

- [1] A. Bartow, A. Kapadia, I. Walker, A novel continuum trunk robot based on contractor muscles, in: Proceedings of the 12th WSEAS International Conference on Signal Processing, Robotics, and Automation, 2013, pp. 181–186.
- [2] Z. Gong, X. Fang, X. Chen, et al., A soft manipulator for efficient delicate grasping in shallow water: modeling, control, and real-world experiments[J], *Int. J. Robot. Res.* 40 (1) (2021) 449–469.
- [3] F. Xu, H. Wang, W. Chen, et al., Visual servoing of a cable-driven soft robot manipulator with shape feature[J], *IEEE Robot. Autom. Lett.* 6 (3) (2021) 4281–4288.
- [4] Y. Zhang, W. Chen, J. Chen, et al., Stiffness analysis of a pneumatic soft manipulator based on bending shape prediction[J], *IEEE Access* 8 (2020) 82227–82241.
- [5] B.A. Jones, I.D. Walker, Kinematics for multisection continuum robots[J], *IEEE Trans. Robot.* 22 (1) (2006) 43–55.
- [6] A. Chawla, C. Frazelle, I. Walker, A comparison of constant curvature forward kinematics for multisection continuum manipulators, in: 2018 Second IEEE International Conference on Robotic Computing (IRC). IEEE, 2018, pp. 217–223.
- [7] Y. Chen, L. Zeng, X. Zhu, et al., Model-based estimation of the gravity-loaded shape and scene depth for a slim 3-actuator continuum robot with monocular visual feedback, in: 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 4416–4421.
- [8] I.S. Godage, G.A. Medrano-Cerda, D.T. Branson, et al., Modal kinematics for multisection continuum arms[J], *Bioinspiration Biomim.* 10 (3) (2015) 035002.
- [9] P.S. Gonithina, A.D. Kapadia, I.S. Godage, et al. Modeling variable curvature parallel continuum robots using euler curves, in: 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 1679–1685.
- [10] W.S. Rone, P. Ben-Tzvi, Continuum robot dynamics utilizing the principle of virtual power[J], *IEEE Trans. Robot.* 30 (1) (2013) 275–287.
- [11] H. Wang, C. Wang, W. Chen, et al., Three-dimensional dynamics for cable-driven soft manipulator[J], *IEEE/ASME Trans. Mechatron.* 22 (1) (2016) 18–28.
- [12] I.S. Godage, G.A. Medrano-Cerda, D.T. Branson, et al., Dynamics for variable length multisection continuum arms[J], *Int. J. Robot. Res.* 35 (6) (2016) 695–722.
- [13] S. Song, Z. Li, H. Yu, et al., Shape reconstruction for wire-driven flexible robots based on Bézier curve and electromagnetic positioning[J], *Mechatronics* 29 (2015) 28–35.
- [14] K.C. Galloway, Y. Chen, E. Templeton, et al., Fiber optic shape sensing for soft robotics[J], *Soft Robot.* 6 (5) (2019) 671–684.
- [15] M.M. Dalvand, S. Nahavandi, R.D. Howe, Fast vision-based catheter 3D reconstruction[J], *Phys. Med. Biol.* 61 (14) (2016) 5128.
- [16] V.K. Chitrakaran, A. Behal, D.M. Dawson, et al., Setpoint regulation of continuum robots using a fixed camera[J], *Robotica* 25 (5) (2007) 581–586.
- [17] J. Li, Y. Sun, H. Su, et al., Marker-based shape estimation of a continuum manipulator using binocular vision and its error compensation, in: 2020 IEEE International Conference on Mechatronics and Automation (ICMA). IEEE, 2020, pp. 1745–1750.
- [18] D.B. Camarillo, K.E. Loewke, C.R. Carlson, et al., Vision based 3-D shape sensing of flexible manipulators, in: 2008 IEEE International Conference on Robotics and Automation. IEEE, 2008, pp. 2940–2947.
- [19] H. Su, C. Yang, G. Ferrigno, et al., Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery[J], *IEEE Robot. Autom. Lett.* 4 (2) (2019) 1447–1453.
- [20] G. Runge, M. Wiese, A. Raatz, FEM-based training of artificial neural networks for modular soft robots, in: 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2017, pp. 385–392.
- [21] I.M. Loutfi, A.H.B. Bouthchouang, A. Melingui, et al., Learning-based approaches for forward kinematic modeling of continuum manipulators[J], *IFAC Pap.* 53 (2) (2020) 9899–9904.
- [22] M. Giorelli, F. Renda, M. Calisti, et al., Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature [J], *IEEE Trans. Robot.* 31 (4) (2015) 823–834.
- [23] A. Melingui, O. Lakhal, B. Daachi, et al., Adaptive neural network control of a compact bionic handling arm[J], *IEEE/ASME Trans. Mechatron.* 20 (6) (2015) 2862–2875.
- [24] H. Su, A. Mariani, S.E. Ovur, et al., Toward teaching by demonstration for robot-assisted minimally invasive surgery[J], *IEEE Trans. Autom. Sci. Eng.* 18 (2) (2021) 484–494.
- [25] R. Wang, S. Wang, S. Du, et al., Real-time soft body 3d proprioception via deep vision-based sensing[J], *IEEE Robot. Autom. Lett.* 5 (2) (2020) 3382–3389.
- [26] X. Li, A.M.H. Tiong, L. Cao, et al., Deep learning for haptic feedback of flexible endoscopic robot without prior knowledge on sheath configuration[J], *Int. J. Mech. Sci.* 163 (2019), 105129.
- [27] H. Su, Y. Hu, H.R. Karimi, et al., Improved recurrent neural network-based manipulator control with remote center of motion constraints: experimental results [J], *Neural Netw.* 131 (2020) 291–299.
- [28] H. Su, W. Qi, Y. Hu, et al., An incremental learning framework for human-like redundancy optimization of anthropomorphic manipulators[J], *IEEE Trans. Ind. Inform.* 18 (3) (2020) 1864–1872.
- [29] A.H.B. Bouthchouang, A. Melingui, J.J.B.M. Ahanda, et al., Forward kinematic modeling of conical-shaped continuum manipulators[J], *Robotica* 39 (10) (2021) 1760–1778.
- [30] O. Lakhal, A. Melingui, R. Merzouki, Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator[J], *IEEE/ASME Trans. Mechatron.* 21 (3) (2015) 1326–1335.
- [31] Qinghua Guan, Jian Sun, Yanjun Liu, et al., Novel bending and helical extensible/contractile pneumatic artificial muscles inspired by elephant trunk[J], *Soft Robot.* 7 (5) (2020) 597–614.
- [32] R.J. Webster III, B.A. Jones, Design and kinematic modeling of constant curvature continuum robots: a review[J], *Int. J. Robot. Res.* 29 (13) (2010) 1661–1683.
- [33] M.D. Grissom, V. Chitrakaran, D. Dienno, et al., Design and experimental testing of the octarm soft robot manipulator[C]//Unmanned Systems Technology VIII, Int. Soc. Opt. Photonics 6230 (2006) 62301F.
- [34] M. Rolf, J.J. Steil, Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 3440–3446.
- [35] S. Zou, Y. Lv, Y. Ma, et al., Design and implement of shape detection for the soft manipulator, in: 2020 39th Chinese Control Conference (CCC). IEEE, 2020, pp. 3972–3977.

**Shuangquan Zou** received the M.Sc. in control Engineering from Harbin Institute of Technology in 2020. He is currently pursuing the Ph.D. degree with Control Science and Engineering, Harbin Institute of Technology. His current research interests include shape detection for soft manipulator, kinematics model, and machine vision.

**Yueyong Lyu** is now an associate research fellow in the Department of Control Science and Engineering, Harbin Institute of Technology, China. He also got his Bachelor, Master and Ph. D degree from Harbin Institute of Technology respectively in 2002, 2008 and 2013. His interests of research mainly focus on spacecraft guidance, navigation and control, especially in spacecraft formation flying, on-orbit service, and so on.

**Jiaming Qi** received the M.Sc. in Integrated Circuit Engineering from Harbin Institute of Technology in 2018. In 2019, he was a visiting PhD student at The Hong Kong Polytechnic University. He is currently pursuing the Ph.D. degree with Control Science and Engineering, Harbin Institute of Technology. His current research interests include data-driven control for soft object manipulation, vision-servoed control, robotics and control theory.

**Guangfu Ma** received the Ph.D. and M.S. degrees in electrical engineering from the Harbin Institute of Technology in 1993 and 1987, respectively. He was with the Harbin Institute of Technology, where he became an Associate Professor in 1992, and a Professor in 1997, where he currently teaches and performs research in optimal control, spacecraft attitude control, and aerospace control systems. He is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology.

**Yanning Guo** is currently a Professor with the Department of Control Science and Engineering, Harbin Institute of Technology. He received the Ph.D. and M.S. degrees from the Harbin Institute of Technology in 2008 and 2012, respectively. His interests of research mainly focus on actuator control distribution, visual navigation and positioning, and so on.