A real-time 3D centerline estimation framework for multi-section soft manipulator based on stereo vision

Shuangquan Zou*, Yueyong Lyu*, Qinghua Guan**, Liwu Liu**, Guangfu Ma*

*School of Astronautics, Harbin Institute of Technology, Harbin 150001

China (Tel: 86-18845610716; e-mail: lvyy@hit.edu.cn).

**Institute of composite materials and structures (Harbin Institute of Technology）, Harbin 150001

Abstract: Soft manipulator is a strong nonlinear system with high uncertainty. Real-time 3D shape estimation is the base for the control and application of soft manipulators. However, it is challenging to realize 3D shape estimation through accurate modeling as the rigid manipulator. To deal with this issue, a real-time 3D centerline estimation framework based on stereo vision is proposed for the multi-section soft manipulator in this work. The contour of the manipulator is segmented accurately from the real-time images captured by the ZED camera using the machine vision method. The contour data is then clustered based on the self-organizing mapping (SOM) algorithm to form a 2D centerline. The linear overdetermined equation established by the camera projection model is figured out to obtain the optimal solution in the sense of least squares, and the 3D reconstruction is completed. In the simulation of the SOM algorithm, the parameters selection, simulation verification, and the comparison of various centerline extraction algorithms are completed. The results show that the SOM algorithm has more advantages to solve this work. Real-time bending experiments are carried out to verify the feasibility and robustness of the proposed framework, and performance evaluation experiments are also performed for accuracy and real-time performance. Compared with other research work, the framework in this work has high accuracy and real-time tracking performance.

Keywords: 3D centerline estimation; contour segmentation; centerline clustering; 3D construction; soft manipulator;

1. INTRODUCTION

Soft manipulator (Chen et al., 2021) is a novel bionic manipulator made of soft materials and can deform continuously along the backbone. The soft materials allow the soft manipulator to grasp the target in a smooth manner and maintain little contact force to achieve compliant capture (Gong et al., 2021). Compared with the rigid structure manipulator (Zhuang et al., 2021), the soft manipulator has natural compliance and safe interaction (Kang et al., 2016).

Although soft manipulators have made significant progress in prototype design and driving strategies, the accuracy performance has been unsatisfactory due to problems such as the nonlinearity of materials and driving coupling between segments during application. Therefore, it is vital to develop an accurate shape sensing system to improve control accuracy (Wang et al., 2020). Shape sensing of soft manipulators has been closely concerned and widely explored by researchers. Constant curvature (Godage et al., 2015) is the simplest and most common for detecting the shape of a soft manipulator. Still, the manipulator cannot deform with constant curvature all time, which leads to low detection accuracy of this methodology. Renda et al. (2012) use Cosserat rod theory combined with a static model for accurate shape estimation. However, in practical applications, the collision model cannot be wholly modeled, and the force sensor cannot accurately measure the contact force. Although the emerging sensing methods for continuous media, such as optical fiber optic sensors (Galloway et al., 2019) and electromagnetic sensors (Song et al., 2015), have good performance in terms of accuracy, it not only restricts the movement of the manipulator, but also have some trouble like high cost and sensitivity to noise.

To sum up, for the soft manipulator, a multivariable and strongly coupled nonlinear system, both the traditional sensing method based on accurate modeling and the burgeoning sensing method with contact measurement have certain limitations. In contrast, the vision-based sensing method (Xu et al., 2021; Zhao et al., 2021) can estimate 3D shape only through image data without affecting the operability of the manipulator. It is a low-cost, high-precision, and non-contact measurement. Hannan et al. (2005) are pioneers in using a monocular camera to extract the center point of the elephant-trunk manipulator for shape determination. But only the plane motion of the manipulator is considered. In stereo vision, installing reference markers (Li et al., 2012; Reilink et al., 2013) is the most common way for shape detection. Nevertheless, the marking point can only be installed on a certain section of the manipulator. When the manipulator moves to an angle that the camera cannot capture, it directly causes the algorithm to fail. Therefore, the unmarked method has more research value. Kumar et al. (2004) propose a self-organizing mapping (SOM) algorithm to model the shape from point cloud for the first time. The results show that the learning-based SOM algorithm is robust in static data clustering and topological mapping. It also seems to be a novel

approach for solving variable and irregular centerline identification of the soft manipulators. Croom et al. (2010) develop a 3D centerline sensing algorithm based on SOM, but this work does not involve real-time research with dynamic targets and even uses a mock-up instead of a soft manipulator for study. Xu et al. (2018) successfully apply the SOM algorithm to real-time centerline recognition of the deformable link model. Not only that, most of the current shape detection systems only accomplish simple plane bending experiments for single-section manipulators or 3D printing models (Camarillo et al., 2008). There is still little research on fast real-time 3D shape reconstruction of the multi-segment manipulator with a complex configuration.

In this work, a real-time 3D centerline estimation framework for the multi-section soft manipulator based on stereo vision is proposed, and the performance verification and comparison are completed through simulations and experiments. Compared with the existing research, the innovations of this work are as follows: 1) SOM algorithm is applied to the real-time 3D centerline estimation of the soft manipulator. 2) The parameter optimization and error compensation of the SOM algorithm ensure the balance between clustering accuracy and time cost. 3) The proposed framework still has high accuracy and real-time performance (compared with other works) for the three-section soft manipulator with complex spatial configuration.

2. OVERVIEW SYSTEM

For the soft manipulator, it is necessary to simultaneously control the end position and the continuous shape during the application process. From the perspective of control theory, real-time monitoring of shape information is an essential guarantee for high-precision control. Therefore, in order to carry out subsequent experimental research and algorithm verification, we built the experimental platform of the control system, as shown in Fig. 1(a).

(a)
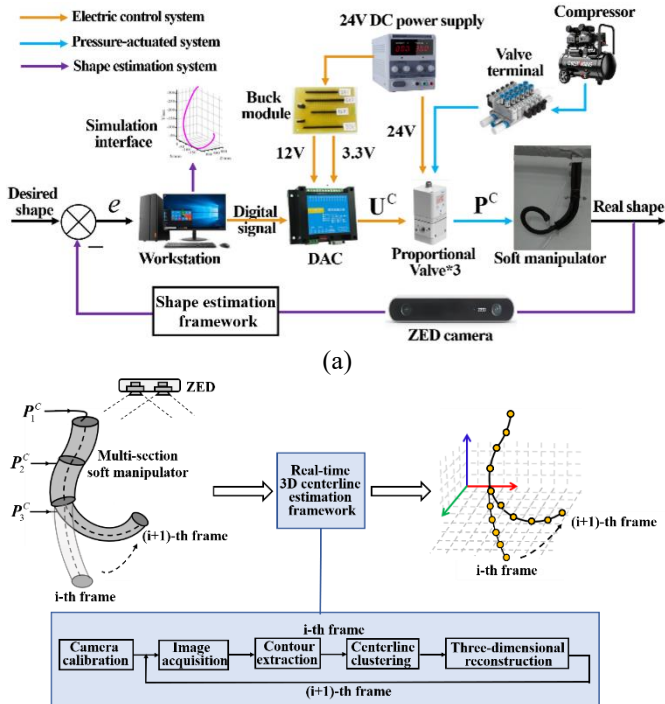
(b)

Fig. 1. Control system overview (a)The block diagram of the proposed system; (b) Schematic diagram of shape estimation framework.

An experimental setup is built to realize the above system. In the pressure-actuated system, the high-pressure source generated by the air compressor is evenly divided after flowing through the valve terminal (Festo, Germany) to form multiple independent pneumatic circuits, which supply air separately for each pneumatic muscle actuators (PMAs). The electric control system can meet the voltage requirements of the components of the experimental platform at all levels and supply power to the whole control system. The DA converter converts the digital signal from the workstation (Intel Xeon Bronze 3104 CPU, 32GB RAM, NVIDIA Quadro P4000) into an analog voltage signal to control the output pressure of the proportional valve (Festo, Germany). Finally, the 3D shape deformation of the soft manipulator is completed by means of pressure driving. As the sensor of the control system, a binocular depth camera (ZED, USA) is placed on the tripod in front of the soft manipulator, and it is guaranteed to cover all the working space of the soft manipulator. The ZED establishes serial communication with the USB3.0 port of the workstation. In addition, all simulations in this work are implemented in Python.

The focus of this work is the shape estimation framework running in the feedback link, which is a real-time framework that can reconstruct the 3D centerline of a multi-segment soft manipulator with high precision without any mechanism modeling. More details of the real-time shape estimation framework are shown in Fig. 1(b).

3. REAL-TIME 3D CENTERLINE ESTIMATION FRAMEWORK

The 3D centerline estimation framework without considering real-time is of little significance, because it cannot be embedded into the closed-loop control system to achieve real-time shape deformation control. The running speed of the framework is opposite to the computational accuracy, so the framework we presented has been trying to find a balance between the two. In the process of framework design, some engineering technologies are added to reduce the amount of unnecessary data calculation and improve real-time performance. Contour segmentation, centerline clustering, and 3D reconstruction are the three core parts of the proposed framework, which will be introduced in detail in each section below.

3.1 Contour segmentation

The purpose of contour segmentation is to accurately segment the soft manipulator from the complex background using machine vision methods. The overall process is divided into the following steps.

Firstly, ZED acquires the left and right image data in real time, and uses the camera internal and external parameters calibrated in the API to correct image distortion so as to obtain a distortion-free image. Secondly, the median filter is adopted

to eliminate the mixed noise in the image. The region of interest (the image data of the soft manipulator) is segmented from the rest of the background according to the HSV color space and marked by binarization. Note that the experimental background is supposed to keep the color single for better image segmentation. Then, the morphological filter is applied to remove isolated pixels to make the contour of the manipulator smoother. Finally, the solution based on the Canny edge detection algorithm (publicly available at OpenCV) is presented for contour segmentation. Using contour data instead of all the image data of the manipulator can greatly reduce the computational cost of the subsequent algorithm. This improved approach was verified in the study of Kumar et al. (2004). The extracted contour data is down-sampled by taking points at equal intervals to thin the contour data and further improve the real-time performance. The results of contour segmentation are shown in Fig. 2.
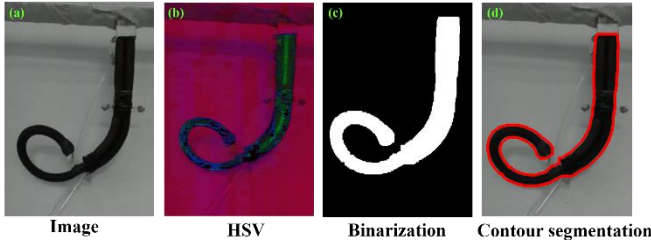

Fig. 2. The results of contour segmentation (the left image).

3.2 Centerline clustering

The multi-segment soft manipulator, whose longitudinal length, is much larger than the radial length is taken as the research object. Therefore, the centerline can well represent the shape of the soft manipulator. In generally, the centerline should be composed of fixed-number, ordered, and equidistant center points. To meet the above requirements, the robust SOM algorithm is proposed to identify the centerline of the image data of the soft manipulator in this section.

SOM is an unsupervised artificial neural network that generates a low-dimensional and discrete mapping by learning the data in the input space. To some extent, it can also be regarded as a dimension reduction algorithm. Here, SOM is used to extract the 2D centerline of the soft manipulator without artificial reference markers. The topological structure of the SOM neural network is shown in Fig. 3.
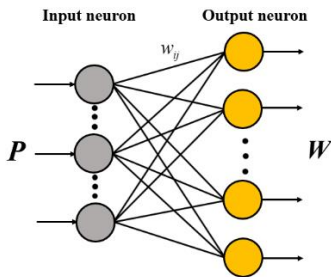

Fig. 3. The topological structure of the SOM neural network.

The number of input neurons m corresponds to the dimension of the input sample. Thanks to the input of SOM is the contour data of the manipulator $\boldsymbol{P} = [\boldsymbol{P}_1, \ldots, \boldsymbol{P}_h] \in \mathbb{R}^{2h}$ here, $m = 2$.

Where h is the number of contour data and $\boldsymbol{P}_k \in \mathbb{R}^2$ is the 2D coordinates of each contour data. As for output neurons, it represents the center points $\boldsymbol{W} = [\boldsymbol{W}_1, \ldots, \boldsymbol{W}_n] \in \mathbb{R}^{2n}$ formed by clustering, where b is the expected number of center points and $\boldsymbol{W}_{L,j} = [u_{L,j}, v_{L,j}] \in \mathbb{R}^2$ (the left image as an example) means the 2D coordinates of each center point. The input and output layer neurons are connected by weight w_{ij}. Referring to the mathematical principle of the SOM algorithm (Xu et al., 2018), the pseudo code of the SOM algorithm is summarized as follows

---

Algorithm 1 SOM Algorithm

1: Initialize the weight vector $\boldsymbol{W}_j(0)$, the neighborhood radius $N \in \mathbb{N}^*$, the initial learning rate $\alpha(0)$, and the training times λ.

2: **loop**

3:   Get current left/right camera frame

4:   Extract the 2D contour point of soft manipulator $\boldsymbol{P}$

5:   **for** $\lambda(0) = 1; \lambda(t) < \lambda; \lambda(t)++$ **do**

6:     **for each** $\boldsymbol{P}_k$ **do**

7:       **for** $j = 1; j < n+1; j++$ **do**

8:         Calculate the Euclidean distance between $\boldsymbol{P}_k$ and each weight vector $\boldsymbol{W}_j$ by the principle of competitive learning, namely

$$\left\| \boldsymbol{P}_k - \boldsymbol{W}_j \right\| = \sqrt{\left(\boldsymbol{P}_k - \boldsymbol{W}_j \right)^{\mathrm{T}} \left(\boldsymbol{P}_k - \boldsymbol{W}_j \right)} \quad (1)$$

9:     **end for**

10:    Find the winner neuron as

$$j^* = \arg\ \min\left(\left\| \boldsymbol{P}_k - \boldsymbol{W}_j \right\| \right) \quad (2)$$

and define the winning neighborhood $N_{j^*}(t)$.

11:     **for all** $j \in N_{j^*}(t)$ **do**

12:       The $\boldsymbol{W}_j$ is updated based on the gradient descent method

$$\boldsymbol{W}_j(t+1) = \boldsymbol{W}_j(t) + \eta(r)\alpha(t)\left[\boldsymbol{P}_k - \boldsymbol{W}_j(t) \right] \quad (3)$$

13:     **end for**

14:    **end for**

15:   **end for**

16: Obtained $\boldsymbol{W}_{L,j}$ and $\boldsymbol{W}_{R,j}$ respectively

17: **end loop**

Remark 1 $N_{j^*}(t)$ is determined with the j^* as the center and N as the neighborhood radius, gradually decreasing over time. In the $N_{j^*}(t)$, the law of neighborhood function $\eta(r)$ is that the farther the topological distance r from j^* is, the smaller $\eta(r)$ is, usually expressed by the Gaussian function.

Remark 2 The learning rate function $\alpha(t)$ is often represented by $1/(1 + \lambda(t)/\lambda)$, which monotonically decreases with the

current training times $\lambda(t)$.

3.3 3D reconstruction

The most common 3D reconstruction methods include numerical methods and analytical methods. The analytical method (Li et al., 2020) is used to calculate the disparity of the target on the left and right images and obtain the depth of the target by the triangulation model. However, this method requires that the left and right cameras must be strictly parallel and the internal parameters are the same, but this is difficult to achieve in the actual production process. In this case, the numerical method is more universal. The linear overdetermined equations are constructed based on the camera projection model, and the least square method is raised to find the optimal solution. The camera projection model and coordinate transformation relationship are shown in Fig. 4. Here, the left camera coordinate $\{O_L\}$ is set as the world coordinate $\{O_W\}$.
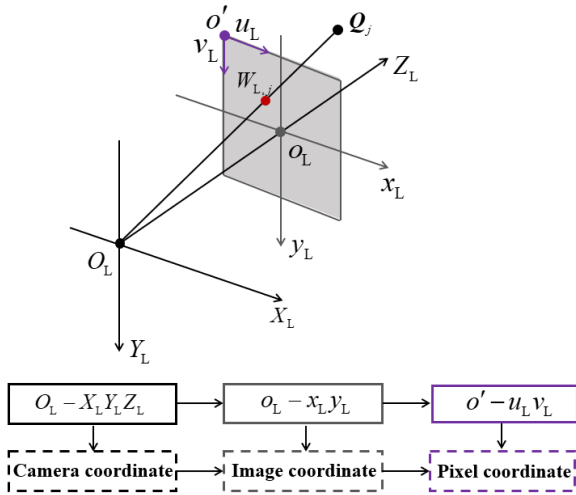

Fig. 4. Camera projection model.

For the internal parameter matrix of the left camera $M_L \in \mathbb{R}^{3\times4}$ (obtained by camera calibration), the camera projection equation is established

$$Z_j \begin{bmatrix} u_{L,j} \\ v_{L,j} \\ 1 \end{bmatrix} = M_L \begin{bmatrix} Q_j \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} m_{L,11} & m_{L,12} & m_{L,13} & m_{L,14} \\ m_{L,21} & m_{L,22} & m_{L,23} & m_{L,24} \\ m_{L,31} & m_{L,32} & m_{L,33} & m_{L,34} \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ Z_j \\ 1 \end{bmatrix}. \quad (4)$$

where the projection of a backbone point $Q_j = [X_j, Y_j, Z_j] \in \mathbb{R}^3$ on the left pixel coordinates is $W_{L,j}$, and their x coordinates and y coordinates are marked as $u_{L,j}$ and $v_{L,j}$, respectively.

Equation (4) can be further simplified to

$$\begin{bmatrix} u_{L,j}m_{L,31} - m_{L,11} & v_{L,j}m_{L,31} - m_{L,21} \\ u_{L,j}m_{L,32} - m_{L,12} & v_{L,j}m_{L,32} - m_{L,22} \\ u_{L,j}m_{L,33} - m_{L,13} & v_{L,j}m_{L,33} - m_{L,23} \end{bmatrix}^T \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}$$

$$= \begin{bmatrix} m_{L,14} - u_{L,j}m_{L,34} \\ m_{L,24} - v_{L,j}m_{L,34} \end{bmatrix}. \quad (5)$$

Similarly, for the internal parameter matrix of the right camera $M_R \in \mathbb{R}^{3\times4}$, a similar equation can also be computed

$$\begin{bmatrix} u_{R,j}m_{R,31} - m_{R,11} & v_{R,j}m_{R,31} - m_{R,21} \\ u_{R,j}m_{R,32} - m_{R,12} & v_{R,j}m_{R,32} - m_{R,22} \\ u_{R,j}m_{R,33} - m_{R,13} & v_{R,j}m_{R,33} - m_{R,23} \end{bmatrix}^T \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}$$

$$= \begin{bmatrix} m_{R,14} - u_{R,j}m_{R,34} \\ m_{R,24} - v_{R,j}m_{R,34} \end{bmatrix}. \quad (6)$$

Simultaneous equations (5) and (6) can be obtained

$$\begin{bmatrix} u_{L,j}m_{L,31} - m_{L,11} & u_{L,j}m_{L,32} - m_{L,12} & u_{L,j}m_{L,33} - m_{L,13} \\ v_{L,j}m_{L,31} - m_{L,21} & v_{L,j}m_{L,32} - m_{L,22} & v_{L,j}m_{L,33} - m_{L,23} \\ u_{R,j}m_{R,31} - m_{R,11} & u_{R,j}m_{R,32} - m_{R,12} & u_{R,j}m_{R,33} - m_{R,13} \\ v_{R,j}m_{R,31} - m_{R,21} & v_{R,j}m_{R,32} - m_{R,22} & v_{R,j}m_{R,33} - m_{R,23} \end{bmatrix} \begin{bmatrix} X_j \\ Y_j \\ Z_j \end{bmatrix}$$

$$= \begin{bmatrix} m_{L,14} - u_{L,j}m_{L,34} \\ m_{L,24} - v_{L,j}m_{L,34} \\ m_{R,14} - u_{R,j}m_{R,34} \\ m_{R,24} - v_{R,j}m_{R,34} \end{bmatrix}. \quad (7)$$

It can be clearly seen that equation (7) is four equations with three unknown variables, which is a linear overdetermined equation. Equation (7) can be summarized as

$$A_j Q_j = B_j \quad (8)$$

To solve the overdetermined equation, the least square method can be employed to find the optimal solution of Q_j. Specific methods include generalized inverse solution or SVD decomposition method, both of which can obtain the optimal solution of Q_j in the sense of least squares. The mature algorithm and the open-source function make the whole calculation process basically have no time expenditure.

Remark 3 $W_{L,j}$ and $W_{R,j}$ clustered by SOM can automatically match as corresponding points without any sorting algorithm because of the ordering guarantee of the SOM.

The backbone vector $Q = [Q_1, ..., Q_n] \in \mathbb{R}^{3n}$ is generated iteratively based on the triangulation model. So far, the 3D centerline estimation of the soft manipulator is completed.

4. NUMERICAL SIMULATIONS

4.1 SOM Parameter selection

There are many optional parameters involved in the SOM algorithm, which are vital to the performance of SOM algorithm. If the parameters are selected randomly (Camarillo et al., 2008), the performance of the SOM algorithm cannot be maximized. In this section, two kinds of errors in the performance of the SOM are defined to perform the parameter optimization selection through error simulation comparison.

The topology error ε_t is described as the proportion of input samples with topological errors in the total samples. The specific content is

$$\varepsilon_t = \frac{1}{r}\sum_{k=1}^{h}\tau(\boldsymbol{P}_k), \quad \tau(\boldsymbol{P}_k) = \begin{cases} 1, & \left|c(j^*) - c(j^{**})\right| \neq 1; \\ 0, & \left|c(j^*) - c(j^{**})\right| = 1. \end{cases} \tag{9}$$

where c represents the serial number of the neuron, and j^{**} refers to the suboptimal neuron. When ε_t exists, it means that the topological relationship of the $\boldsymbol{W}_j$ is chaotic, which will directly lead to the failure of 3D reconstruction. Therefore, ε_t need to be eliminated.

Quantization error ε_q evaluates the accuracy of SOM by calculating the average Euclidean distance between each $\boldsymbol{P}_k$ and corresponding $\boldsymbol{W}_{j^*}$. Although ε_q cannot be eliminated as 0, it will gradually converge in the training process, expressed as

$$\varepsilon_q = \frac{1}{r}\sum_{k=1}^{h}\left\|\boldsymbol{P}_k - \boldsymbol{W}_{j^*}\right\|. \tag{10}$$

Next, four error simulations are carried out to design the SOM parameters, including n, N, $\alpha(0)$, and λ.

(1) Simulation case 1: We select n by judging whether $\varepsilon_t = 0$ because other parameters of the SOM will not affect the topological mapping of the data. To avoid topology error caused by insufficient training, set sufficient training times and conduct simulations, as shown in Fig. 5(a). The results imply that when $n \leq 30$, $\varepsilon_t = 0$. The right side also gives the corresponding centerline clustering in the case of $n = 50$ and $n = 70$. It clearly shows that the topological relationship confusion has appeared at this time.

(2) Simulation case 2: Choose $n = 15, 20, 25, 30$ and further study the influence of training times on the convergence of topology error. From Fig. 5(b), n is positively related to λ. For the consideration of the balance between real-time performance and accuracy, choose $n = 20$.

(3) Simulation case 3: It can be proved that N and $\alpha(0)$ determine ε_q, while λ only affects the convergence speed.

Set $N = 2, 4, 6, 8$, explore the relationship between $\alpha(0)$ and ε_q (still set sufficient λ). The simulation results are shown in Fig. 5(c). Under any $\alpha(0)$, the smaller N is, the smaller ε_q is. It can be explained that if N is too large, it will promote the $\boldsymbol{W}_j$ clustering to the same point, which deviates from the real distribution of the samples. Hence, let $N = 2$. Moreover, no matter $\alpha(0)$ is excessively large or small, the result is not very ideal. Excessive $\alpha(0)$ can cause inaccurate learning, and too small $\alpha(0)$ will fall into the local learning. From Fig.5(c), $\alpha = 0.2$ is the most appropriate choice.
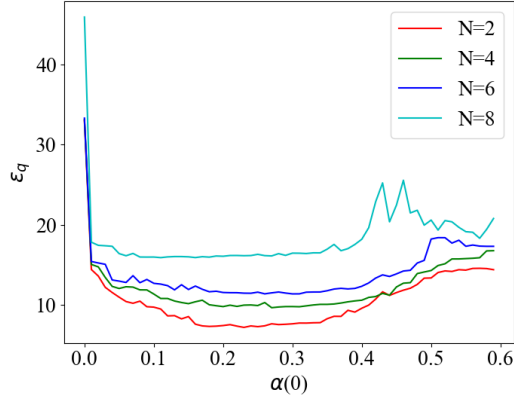
(4) Simulation case 4: Then, shorten the learning rate range ($0 \leq \alpha(0) \leq 0.3$), and investigate the influence of λ on the convergence speed of ε_q. It can be seen from Fig. 5(d) that although $\lambda = 15, 20, 25$ converges consistently in the case of $\alpha = 0.2$, the increase of λ will prolong the calculation time, which is unfavorable for the real-time design. So $\lambda = 15$ is a better choice.
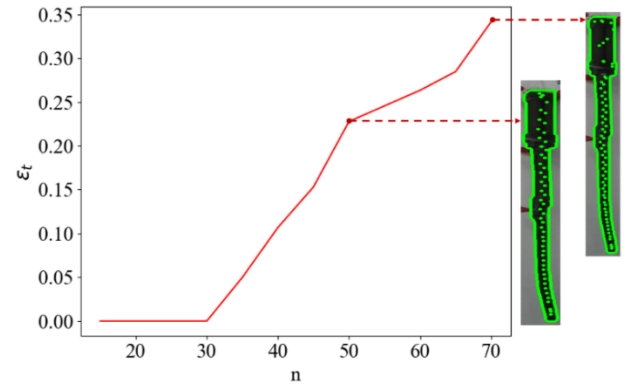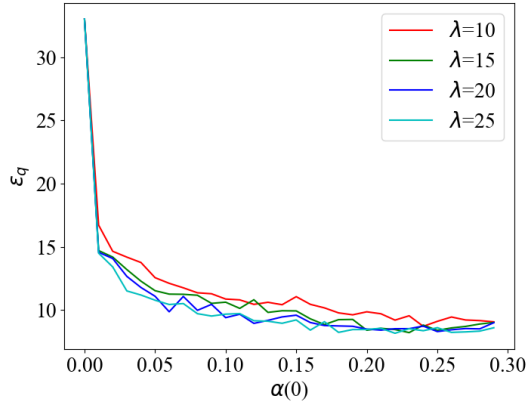

(a)
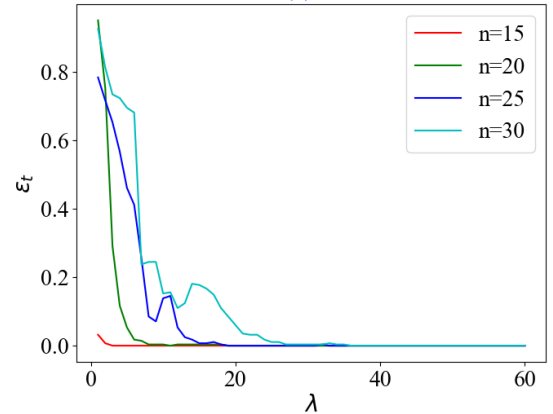
(b)

(c)

(d)

Fig. 5. Simulations for SOM error analysis. (a) Simulation case 1; (b) Simulation case 2; (c) Simulation case 3; (d) Simulation case 4.

Ultimately, the SOM parameters are selected in Table 1, combined with the simulation results.

Table 1. SOM parameter design

n	N	$\alpha(0)$	λ
20	2	0.2	15

4.2 SOM simulation results

In this section, the simulation verification of the SOM algorithm is carried out for the initial posture of the soft manipulator. SOM initialization includes linear initialization and random initialization, and linear initialization can capture the real initial shape of the soft manipulator in a shorter time. In Fig. 6(a), the dynamic process of SOM training is expressed by gradient red. With the increase of λ, $W_{L,j}$ gradually moves from the initial state $W_{L,j}(0)$ (described by blue) to the position of the centerline. Finally, the left and right image simulation results of the SOM algorithm for the initial attitude of the soft manipulator are shown in Fig. 6(b). Additionally, starting from the first frame, the SOM training result of the previous frame is treated as the initial weight vector for the next frame, thereby further improving the tracking performance of the SOM algorithm.

It is worth noting that there is no y-axis offset after image correction when a certain point in physical space is projected on the left and right images. However, the SOM training on the left and right images is performed separately, so there is no guarantee that the corresponding points do not have a y-axis offset. This offset on the y-axis is named as the misalignment error ε_w

$$\varepsilon_w = \frac{1}{n}\sum_{j=1}^{n}\left|v_{L,j} - v_{R,j}\right|. \tag{11}$$

To completely compensate for this error, taking $W_{L,j}$ as a reference, cubic spline interpolation is figured out on $W_{R,j}$. In the generated interpolation, the new data points that satisfy $v_{R,j}^{*} = v_{L,j}$ are found as the compensated right weight vector $W_{R,j}^{*}$. The SOM simulation results before and after error compensation are shown in Fig. 6(c). The numerical simulation of the misalignment error is shown in Fig. 6(d). After calculation, the ε_w for the initial posture of the soft manipulator is 1.53.
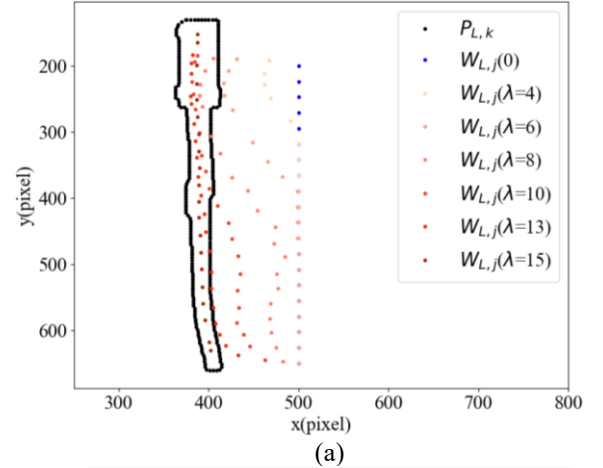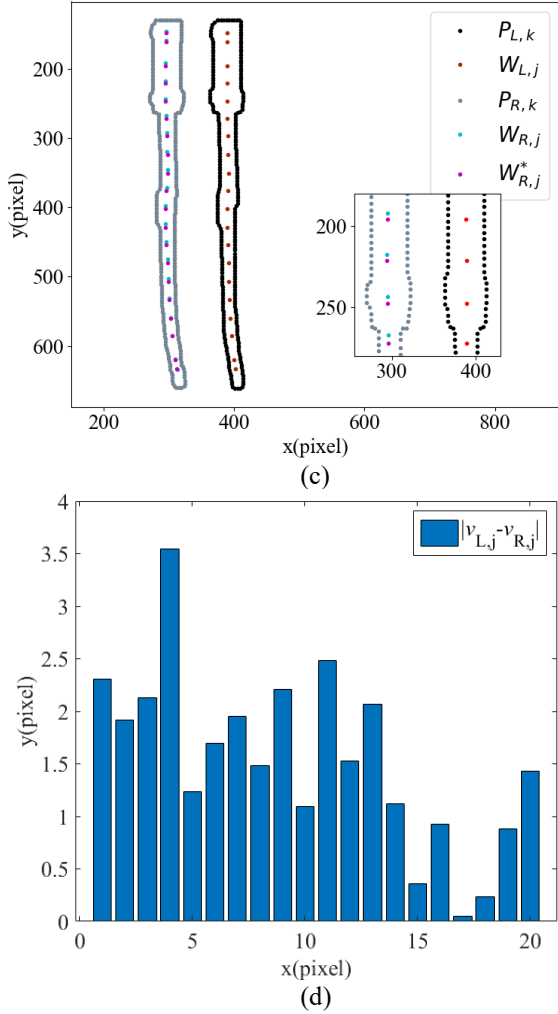

(a)
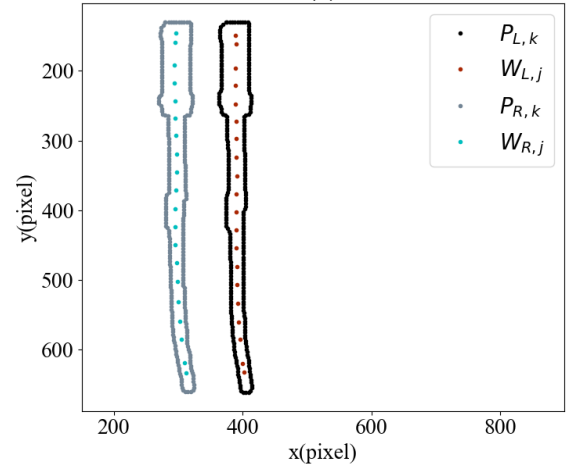
(b)

Fig. 6. SOM algorithm simulation results. (a) Linear initialization and dynamic training process (left image); (b) Simulation results of the SOM algorithm; (c) The final SOM simulation results after error compensation; (d) Numerical simulation for the misalignment error.

4.3 Centerline extraction comparison

In order to clarify the superiority of the SOM algorithm in centerline extraction, the other three centerline extraction algorithms in machine learning, machine vision, and other fields are proposed for comparative study.

Firstly, the competitive learning algorithm, a common unsupervised learning strategy, is introduced, which is the initial version of the SOM algorithm. The difference between it and SOM algorithm is that the competitive learning algorithm takes WTA (Winner-Take-All) as the principle and only adjusts the weights of the winning neurons, while SOM additionally defines the concept of the winning neighborhood and adjusts the weights of the neurons in the whole winning neighborhood according to the distance. The pseudocode is summarized in Algorithm 2.

Algorithm 2 Competitive learning algorithm

1: Initialize the weight vector $W_j(0)$, the learning rate $\alpha \in (0,1)$, the training times λ, and input 2D contour P.

2: **for** $\lambda(0)=1$; $\lambda(t)<\lambda$; $\lambda(t)++$ **do**

3: **for each** P_k **do**

4:   **for** $j=1$; $j<n+1$; $j++$ **do**

5:    Calculate the Euclidean distance between P_k and each weight vector W_j by the principle of competitive learning, namely

$$\left\| P_k - W_j \right\| = \sqrt{\left(P_k - W_j \right)^{\mathrm{T}} \left(P_k - W_j \right)}$$

6:   **end for**

7: Find the winner neuron as

$$j^* = \arg\ \min\left(\left\| P_k - W_j \right\| \right)$$

8:    Only the winning neurons W_{j^*} is updated based on the gradient descent method

$$W_{j^*}(t+1) = W_{j^*}(t) + \alpha \left[P_k - W_{j^*}(t) \right]$$

9: **end for**

10: **end for**

11: Obtained $W_{\mathrm{L},j}$ and $W_{\mathrm{R},j}$ respectively

---

Then, the K-Means++ algorithm is an improved version of K-means, the most classical clustering algorithm in machine learning. This improvement is mainly reflected in the fact that the selection of the initial point is no longer randomly selected like K-means but calculates the shortest distance between each input sample and the known initial point. The greater the distance, the greater the probability of being selected as the initial point. Use route wheel selection to select the next initial point. This method ensures that the distance between the initial points is as far as possible, which can further reduce the clustering error. After the initial point is selected, the subsequent calculation steps are entirely consistent with K-means. The pseudocode is given in Algorithm 3. For the bending shape of the soft manipulator, the experimental results of the above three machine learning algorithms are shown in Fig. 7.

Algorithm 3 K-Means++ algorithm

1: Select the center points number n and iterations number $l=15$, and input 2D contour P.

2: Initialize a sample $W_1(0)$ randomly.

3: **for** $j=1$; $j<n$; $j++$ **do**

4: **for each** P_k **do**

5:    Calculate the shortest distance

$$D_k = \min_{r \in [1,j]} \left\| P_k - W_r(0) \right\|$$

6:    Calculate the probability of being selected as the initial point

$$\sigma_k = \frac{D_k^2}{\sum\limits_{k=1}^{h} D_k^2}$$

7: **end for**

8: Select the next initial point $W_{j+1}(0)$ by referring to the route wheel selection.

9: **end for**

10: Complete initialization $W(0)$

11: Define the set $C = \{C_1,...,C_n\} = \varnothing$

12: **for** $l = 1;\ l < 16;\ l++$ **do**

13:   **for each** P_k **do**

14:     Calculate $j' = \arg \min_{j \in [1,\,n]} \left(\left\| P_k - W_j \right\| \right)$

        Assign $C_{j'} = C_{j'} \cup P_k$.

15:   **end for**

16:   **for** $j = 1;\ j < n+1;\ j++$ **do**

17:    Update $W_j = \sum_{P_k \in C_{j'}} P_k \left/ \left| C_{j'} \right| \right.$ using all P_k belonging to

    $C_{j'}$.

18:   **end for**

19: **end for**

20: Output the center points W
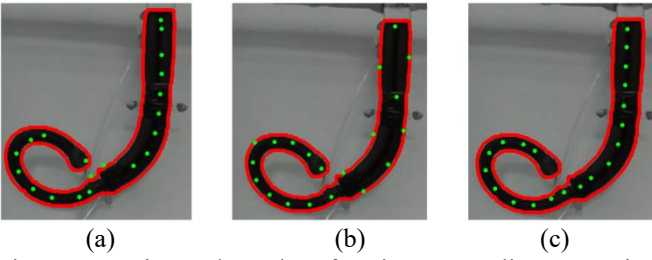

(a)   (b)   (c)

Fig. 7. Experimental results of various centerline extraction algorithms. (a) Competitive learning; (b) K-Means++; (c) SOM.

What's more, thinning (Anton et al., 2017) is also a popular machine vision method to extract the centerline. It can thin the connected component pixel width of the binary image of the soft manipulator to one pixel for backbone extraction and topology representation. The algorithm steps are shown in Fig. 8(a). After that, the desired number of center points is solved by taking points at equal intervals on the obtained centerline, and the center points extraction of the manipulator is completed. However, when the manipulator is not monotonically deformed along a certain axis, the center points obtained by this method have a chaotic topology relationship (see Fig. 8(b)).
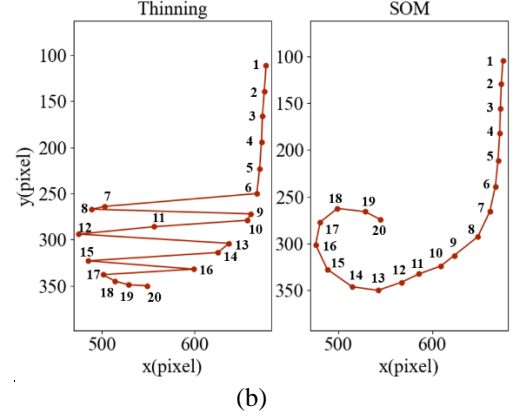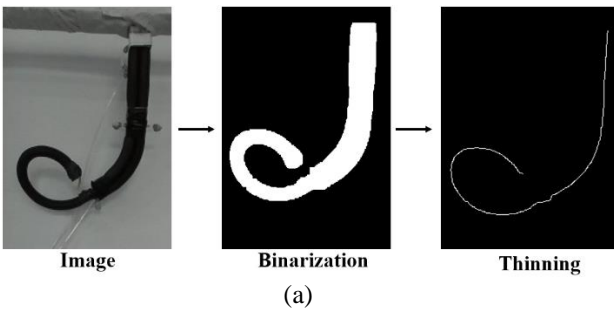

(a)

(b)

Fig. 8. The effect of the thinning algorithm. (a) Steps of the thinning algorithm; (b) The topological relationship of the thinning and SOM. The center points obtained by the two algorithms are connected in order.

To sum up, the results indicate that the competitive learning, K-means++, and thinning algorithm are not suitable for this research work. Competitive learning algorithm cannot accurately learn complex input samples because only the winning neuron is adjusted each time. K-means++ is too sensitive to initial conditions, which greatly affects the clustering results. The single iterative calculation process for the average value makes it have better results only for the input data with the circular distribution. The thinning algorithm requires an additional sorting algorithm to get the ordered center point. The SOM algorithm can not only intelligently learn complex input sample data, but also output the ordered center points. Therefore, the SOM algorithm is applied to the framework of this work for centerline extraction.

5. EXPERIMENTAL RESULTS

In this section, a real-time bending deformation experiment and comparative verification experiments are conducted, respectively, to evaluate the performance of the proposed framework by error analysis, including accuracy and real-time performance.

5.1 Setup

To further validate the effectiveness of the proposed framework, the experimental platform with the soft manipulator is established to conduct the experimental study. The details of the setup composition are shown in Fig. 9(a). In this work, the three-section pneumatic soft manipulator (Guan et al., 2020) is taken as the prototype to verify the framework. Each section of the prototype has PMAs with different bending modes, depending on the placement of the internal elastic frame, as shown in Fig.9(b). In the first section, four bending contractile PMAs (the braid angle is below 54.74°) symmetrically distributed in parallel are fixed on the baseplate, which can bear the maximum load. The second section is composed of two bending extensile PMAs (the braid angle is larger than 54.74°) symmetrically distributed in parallel, which is considered to provide a larger range of motion. The third section adopts a helical extensile PMA that produces both bending and rotational motions to facilitate grasping the target

object. Each section of the prototype is connected in series by a 3D printed baseplate, and the maximum aspect ratio can reach 13.7.
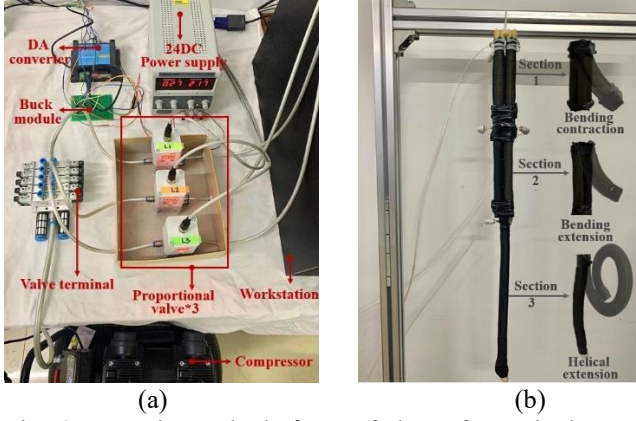
(a)   (b)

Fig. 9. Experimental platform of the soft manipulator. (a) Detail of the experimental setup; (b) The soft manipulator used in this work.

5.2 Real-time bending deformation experiment

A real-time bending deformation experiment is designed to verify the feasibility and robustness of the proposed framework. The voltage driver of the proportional valve is programmed to continuously increase the pressure filled into each section of the soft manipulator from 0 bar to 2.5 bar in fixed steps of 0.05 bar, thereby gradually deforming the manipulator to the maximum bending posture. ZED camera is opened to capture the image, and the resolution is 1280 × 720. At the same time, the shape estimation algorithm we proposed is run to track the 3D posture of the soft manipulator in real time. The whole dynamic bending process lasted for 58s, and 690 groups of data were generated. The contour segmentation results during deformation are shown in Fig. 10(a). The selected data at equal time intervals are printed on the frame (20 sets of data in total), and the initial and end posture are marked in red. The relative plane centerline extraction results based on the SOM algorithm $\boldsymbol{W}_{\mathrm{L}}$ and final 3D reconstruction results $\boldsymbol{Q}$ are shown in Fig. 10(b) and Fig. 10(c), respectively. For better description in the continuum form, the 6th-degree polynomial fitting is developed for each pose to form a smooth 3D space curve.
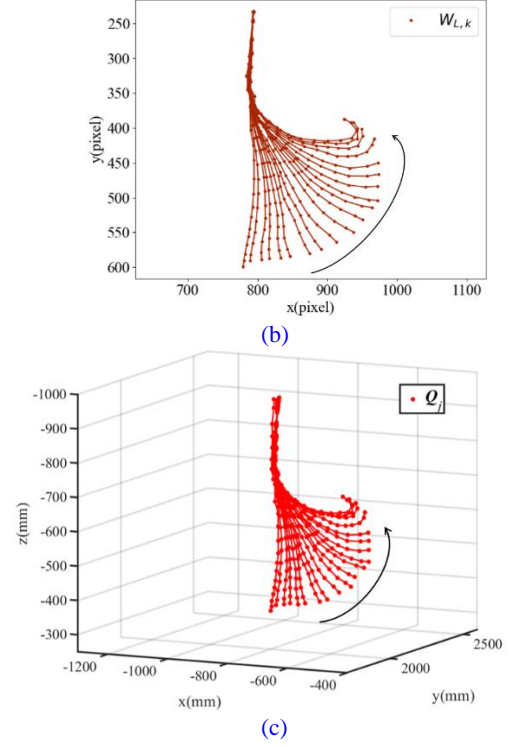
(a)

(b)

(c)
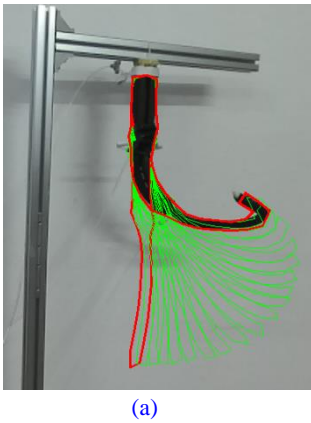
Fig. 1. Experimental results. (a) Contour segmentation during motion; (b) Plane centerline extraction results (c) Final 3D centerline estimation results.

It should be pointed out that the first section manipulator is bent backward, orthogonal to the bending direction of the second section. Furthermore, the bending of the third section also appears as 3D torsion, but these actuation characteristics are not apparent from the above shooting angle. The framework we proposed has strong robustness to sense a variety of complex 3D postures of the soft manipulator in real time.

5.3 Performance verification and error analysis

Next, the accuracy of the proposed framework is evaluated. As a depth camera, ZED's built-in depth detection function can directly obtain 3D coordinates from the 2D left image coordinates of the target, that is, $\boldsymbol{Q}'_j = \left[X'_j,\ Y'_j,\ Z'_j \right] = g\left(\left[u_{\mathrm{L},j},\ v_{\mathrm{L},j} \right] \right)$. For the result of the same plane SOM algorithm in each frame, the depth detection function is called to obtain $\boldsymbol{Q}'$ directly and compare it with the $\boldsymbol{Q}$ calculated by our framework. The 3D centerline comparison for the selected pose is shown in Fig. 11.

Remark 4 The measurement error of the ZED camera is ignored to default $\boldsymbol{Q}'$ as the true value.
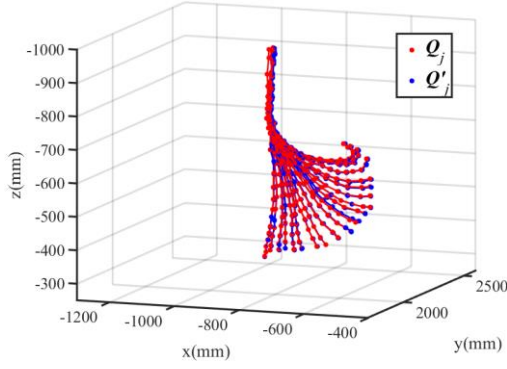
Fig. 2. The 3D centerline comparison for the selected pose.

Obviously, Q is able to continuously approach to the Q' with high accuracy. The following detailed error analysis will quantify the accuracy during the dynamic bending process. Calculate the root mean square error (RMSE)

$$e_{RMSE} = \left(\sum_{j=1}^{n} \left\| Q_j - Q'_j \right\|_j^2 / n \right)^{1/2}$$

between the experimental

results Q and the true value Q' on each axis per frame, and the numerical error simulations before and after misalignment error compensation are shown in Fig. 12. In the process of dynamic deformation, the pose of the prototype at four times is recorded and marked at the corresponding time. It can be observed that RMSE is significantly reduced in each axis via misalignment error compensation. The z-axis error of each frame is the largest compared with other axes whether error compensation or not, because the depth is generated by employing the least squares method to solve the optimal solution of the system of linear overdetermined equations. Overall, in the whole process of dynamic deformation, the error fluctuation range is relatively stable, which also suggests that the proposed framework has strong robustness.
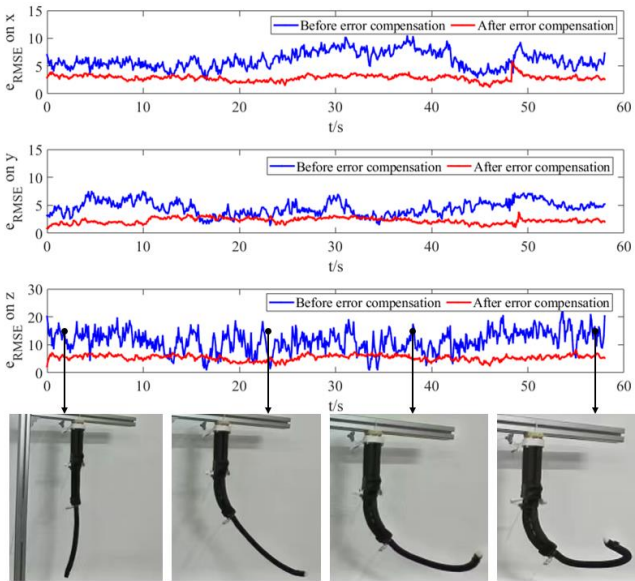

Fig. 3. Error simulations on x, y, z axes during the motion.

Based on the above error data, the RMSE on each dimension during the entire motion (After error compensation) is further

calculated as published in Table 2. Referring to the size of the entire manipulator (total length of 685mm), the accuracy is still within an acceptable range. The generation of errors can be analyzed for the following two reasons: 1) ZED itself has an inevitable measurement error (<1%) for depth detection, especially the stereo camera is sensitive to illumination changes, resulting in a deviation between Q' and the real value; 2) In the 3D reconstruction model, the numerical method can only find the optimal solution but can never get the exact solution, which also produces certain errors;

Table 2. RMSE during the entire motion

x -axis	y -axis	z -axis
2.9178	2.3287	5.5122

In terms of real-time performance, the running time can be measured by setting the getTickCount function (supported by OpenCV) in the designed program. The fps (frames per second) during bending deformation is recorded in real time, as shown in Fig. 13, which indicates that our proposed algorithm framework can run stably at around 10 fps and perform well in real-time performance. Note that compensating for the misalignment error reduces the frame rate slightly but is not evident. Among them, there are two parts that mainly occupy time: 1) The SOM algorithm is executed independently for the left and right images per frame, which takes about 0.061s. 2) Owing to the slow real-time update of Matplotlib-3D, the 3D visualization program runs 0.037s, and other 3D visualization modules will be considered to replace it in the future. The factors affecting real-time performance also include external factors such as processor performance and camera resolution. But in fact, the current fps of the framework is enough to provide shape information for the controller in real time.
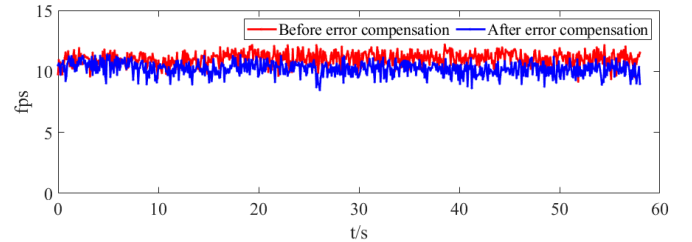

Fig. 13. The real-time performance simulation curve.

In the end, the proposed framework is compared with five vision-based shape sensing algorithms for the soft manipulator in the current works. Table 3. lists that the performance of our framework has been improved in all aspects. The real-time 3D centerline estimation video is available in the supplementary file to further demonstrate it.

Table 3. Comparison with other research work

Algorithm	Structure configuration	Maximum error / manipulator length	Real-time performance /fps
Our work	Three sections	5.5/680 =0.81%	10
AlBeladi et	Three sections	28.2/287	Not designed

		=9.8%	
Manakov et al. (2021)	Two sections	0.56/80 =0.7%	1.43
Lai et al. (2020)	Two sections	20.15/310 =6.5%	Not designed
Vandini et al. (2014)	One section	2.98/60 =4.9%	2.78
Camarillo et al. (2008)	Alternative model	7.84/160 =4.9%	3-4

6. CONCLUSION

In this work, a real-time 3D centerline estimation framework is proposed to reconstruct the spatial posture for the three-section soft manipulator, and the algorithm verification and the performance evaluation are completed. The framework we designed integrates machine vision, machine learning, and real-time computing. The main conclusions are as follows:

(1) For the plane centerline extraction of the soft manipulator, the clustering process of the SOM algorithm is more intelligent than other algorithms, and it has stronger robustness and effectiveness for the complex centerline extraction. In addition, the SOM algorithm also has order preserving for output, which is more suitable to settle this work.

(2) Experiments show that the proposed framework can stably estimate the centerline in real time during the whole dynamic bending process.

(3) The accuracy and real-time performance are verified. The results confirm that the maximum RMSE is 5.5122, and the running speed can achieve 10fps. It is worth mentioning that although the misalignment error compensation makes the real-time performance of the framework decline slightly, the accuracy has been greatly improved, which proves that the compensation is meaningful.

(4) Compared with the visual shape detection algorithms in other works, our framework still has better accuracy and real-time performance for the complex three-section soft manipulator.

ACKNOWLEDGEMENT

REFERENCES

Albeladi, A., Krishnan, G., Belabbas, M. A. and Hutchinson, S. (2021), "Vision-Based Shape Reconstruction of Soft Continuum Arms Using a Geometric Strain Parametrization", *In 2021 IEEE International Conference on Robotics and Automation*, IEEE, pp. 11753-11759.

Anton, F. D., Borangiu, T., Anton, S., Raileanu, S. (2017), "Cloud Robot Vision Services Extend High-Performance Computing Capabilities of Robot Systems", *In 2017 IEEE International Conference on Robotics in Alpe-Adria Danube Region, IEEE*, pp. 317-327.

Camarillo, D. B., Loewke, K. E., Carlson, C. R. and Salisbury, J. K. (2008), "Vision based 3-D shape sensing of flexible manipulators", *In 2008 IEEE International Conference on Robotics and Automation*, IEEE, pp. 2940-2947.

Chen, Y., Li, W. and Gong, Y. (2021), "Static modeling and analysisof soft manipulator considering environment contact based on segmented constant curvature method", *Industrial Robot*, Vol. 48 No. 2, pp. 233-246

Croom, J. M., Rucker, D. C., Romano, J. M. and Webster, R. J. (2010,), "Visual sensing of continuum robot shape using self-organizing maps", *In 2010 IEEE International Conference on Robotics and Automation*, IEEE, pp. 4591-4596.

Galloway, K. C., Chen, Y., Templeton, E., Rife, B., Godage, I. S. and Barth, E. J. (2019), "Fiber optic shape sensing for soft robotics", *Soft robotics*, Vol.6 No. 5, pp. 671-684.

Godage, I. S., Medrano-Cerda, G. A., Branson, D. T., Guglielmino, E. and Caldwell, D. G. (2015), "Modal kinematics for multisection continuum arms", *Bioinspiration & biomimetics*, Vol.10 No. 3, p. 035002.

Gong, Z., Fang, X., Chen, X., Cheng, J., Xie, Z., Liu, J., Chen, B., Yang, H., Kong, S., Hao, Y., Wang, T., Yu, J. and Wen, L. (2021), "A soft manipulator for efficient delicate grasping in shallow water: Modeling, control, and real-world experiments", *The International Journal of Robotics Research*, Vol.40 No. 1, pp. 449-469.

Guan, Q., Sun, J., Liu, Y., Wereley, N. M. and Leng, J. (2020), "Novel bending and helical extensile/contractile pneumatic artificial muscles inspired by elephant trunk", *Soft robotics*, Vol.7 No. 5, pp. 597-614.

Hannan, M. W. and Walker, I. D. (2005), "Real-time shape estimation for continuum robots using vision", *Robotica*, Vol.23 No. 5, pp. 645-651.

Kang, R., Guglielmino, E., Zullo, L., Branson, D. T., Godage, I. and Caldwell, D. G. (2016), "Embodiment design of soft continuum robots", *Advances in Mechanical Engineering*, Vol. 8 No. 4, pp. 1-13

Kumar, G. S., Kalra, P. K. and Dhande, S. G. (2004), "Curve and surface reconstruction from points: an approach based on self-organizing maps", *Applied Soft Computing*, Vol.5 No. 1, pp. 55-66.

Lai J., Huang K., Lu B. and Chu, H. K. (2020), "Toward vision-based adaptive configuring of a bidirectional two-segment soft continuum manipulator", *In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, pp. 934-939.

Li, J., Sun, Y., Su, H., Zhang, G. and Shi, C. (2020), "Marker-Based Shape Estimation of a Continuum Manipulator Using Binocular Vision and Its Error Compensation", *In*

2020 IEEE International Conference on Mechatronics and Automation, IEEE, pp. 1745-1750.

Li, J., Wang, W. and Zhang, M. J. (2012), "3D reconstruction method combined by binocular stereoscopic vision and coded-structured light", *Journal of Computer Applications*, Vol.32 No. s2, pp. 154-158.

Manakov, R. A., Kolpashchikov, D. Y., Danilov, V. V., Laptev, N. V., Skirnevskiy, I. P. and Gerget, O. M. (2021), "Visual shape and position sensing algorithm for a continuum robot", *In IOP Conference Series: Materials Science and Engineering*, IOP Publishing, p. 012066.

Renda, F., Cianchetti, M., Giorelli, M., Arienti, A. and Laschi, C (2012), "A 3D steady-state model of a tendon-driven continuum soft manipulator inspired by the octopus arm", *Bioinspiration & biomimetics*, Vol. 7 No.2, pp.025006.

Reilink, R., Stramigioli, S. and Misra, S. (2013), "3D position estimation of flexible instruments: marker-less and marker-based methods", *International journal of computer assisted radiology and surgery*, Vol.8 No. 3, pp. 407-417.

Song, S., Li, Z., Yu, H. and Ren, H. (2015), "Shape reconstruction for wire-driven flexible robots based on Bézier curve and electromagnetic positioning", *Mechatronics*, Vol.29, pp. 28-35.

Vandini, A., Salerno, A., Payne, C. J. and Yang, G. Z. (2014). "Vision-based motion control of a flexible robot for surgical applications", In 2014 IEEE International Conference on Robotics and Automation, IEEE, pp. 6205-6211.

Wang, Y., Wang, H., Liu, Z. and Chen, W. (2020), "Visual Servo-Collision Avoidance Hybrid Task by Considering Detection and Localization of Contact for a Soft Manipulator", *IEEE/ASME Transactions on Mechatronics*, Vol.25 No. 3, pp.1310-1321.

Xu, F., Wang, H., Chen, W. and Miao, Y. (2021), "Visual servoing of a cable-driven soft robot manipulator with shape feature", *IEEE Robotics and Automation Letters,* Vol.6 No. 3, pp. 4281-4288.

Xu, S., Li, G., Song, D., Sun, L. and Liu, J. (2018), "Real-time shape recognition of a deformable link by using self-organizing map", *In 2018 IEEE 14th International Conference on Automation Science and Engineering*, IEEE, pp. 586-591.

Zhao, Q., Lai, J., Huang, K., Hu, X. and Chu, H. K. (2021), "Shape Estimation and Control of a Soft Continuum Robot Under External Payloads", *IEEE/ASME Transactions on Mechatronics.*

Zhuang, J., Li, R., Cao, C., Gao, Y., Wang, K. and Wang, F. (2021), "Simultaneous identification of joint stiffness, kinematic and hand-eye parameters of measurement system integrated with serial robot and 3D camera", *Industrial Robot*, Vol. 48 No. 4, pp. 484-493.