

RESEARCH ARTICLE

# Image-based anti-interference robotic Chinese character writing system

Xian Li<sup>1</sup> , Chenguang Yang<sup>1</sup> , Sheng Xu<sup>2</sup> and Yongsheng Ou<sup>3</sup> 

<sup>1</sup>College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, <sup>2</sup>Guangdong Provincial Key Lab of Robotics and Intelligent System, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, and <sup>3</sup>Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China

**Corresponding author:** Chenguang Yang; Email: [cyang@ieee.org](mailto:cyang@ieee.org)

**Received:** 23 March 2023; **Revised:** 20 November 2023; **Accepted:** 12 December 2023;

**First published online:** 22 January 2024

**Keywords:** computer vision; force control; control of robotic systems; visual servoing; dynamic movement primitives

## Abstract

This article designs a robotic Chinese character writing system that can resist random human interference. Firstly, an innovative stroke extraction method of Chinese characters was devised. A basic Chinese character stroke extraction method based on cumulative direction vectors is used to extract the components that make up the strokes of Chinese characters. The components are then stitched together into strokes based on the sequential base stroke joining method. To enable the robot to imitate handwriting Chinese character skills, we utilised stroke information as the demonstration and modelled the skills using dynamic movement primitives (DMPs). To suppress random human interference, this article combines improved DMPs and conductance control to adjust robot trajectories based on real-time visual measurements. The experimental results show that the proposed method can accurately extract the strokes of most Chinese characters. The designed trajectory adjustment method offers better smoothness and robustness than direct rotating and translating curves. The robot is able to adjust its posture and trajectory in real time to eliminate the negative impacts of human interference.

## 1. Introduction

Today, robots are used in a wide range of applications. Robots show great advantages in industrial areas where repetitive work is performed because of their low cost, high efficiency and low scrap rate [1]. But in areas where motion trajectories do not repeat, robots have difficulty completing tasks. One of the classic tasks is handwriting. Handwriting actions are unavoidable in human life and play an important role in scenarios such as teaching and businesses both requiring signatures [2, 3]. English words are made up of 26 letters arranged in a row, but Chinese characters are two-dimensional pictures made up of Chinese strokes written in specific positions superimposed on each other [4]. How to encode Chinese characters is a challenge. As a result, the robotic writing of Chinese characters has also been studied by numerous researchers. At the same time, the structure of the strokes of Chinese characters has evolved over millennia to conform to the human mind and is the scientifically best way to segment Chinese characters. In other words, the segmentation of Chinese character strokes can also be applied to tasks that require the segmentation of superimposed trajectory images into executable trajectories, such as the grinding and cleaning of complex recesses in workpieces. At the same time, writing is a delicate manipulation task that would be difficult to accomplish if interference were encountered. Therefore, there is an urgent need to design an interference-resistant robotic Chinese character writing system for diverse applications.

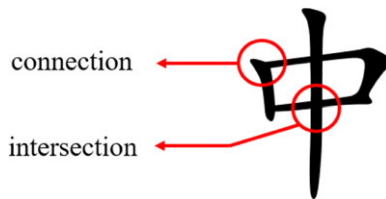
The first step in imitating human writing is to disassemble Chinese characters into Chinese strokes. Currently, three methods are widely applied for Chinese character extraction, i.e., (a) computer font

reproduction, (b) imitation of human writing trajectories and (c) Chinese character image decomposition. Computer font reproduction involves using the stroke information that comes with a standard font library [5–8] and handling of word posters [9–13], but these methods are database dependent and cannot imitate the individual writing of a particular person. Imitation of human writing trajectories includes recording the trajectory of the pen tip when a person is writing [14, 15]. The motion sensing input devices are utilised to capture the gestures and trajectory of the human hand [4], and the physical contact demonstrations [16]. However, these methods rely on human teaching and cannot break down Chinese strokes through pictures. Chinese character image decomposition means that it is possible to analyse the strokes of Chinese characters directly through pictures without relying on word library and human teaching. Chinese character image decomposition, the most difficult but widely used method, has been studied by many researchers. Consequently, many methods were developed, such as corner detection algorithm [17, 18], point to boundary orientation distance of one triangular mesh [19], character library template matching segmentation [20, 21], B-spline curve matching [22], extraction of strokes using ambiguous zone information [18], extraction of strokes using optimum paths [11], stroke speed feature and stroke vector feature segmentation [23], etc. However, these methods require a lot of computation. There is still a need for a simple picture-based method to extract strokes from Chinese characters.

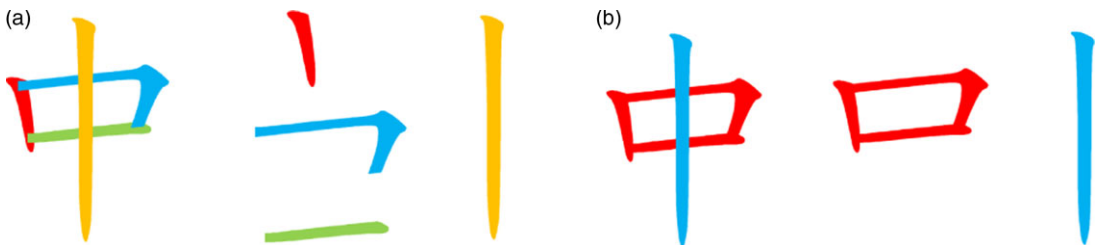
The robot can write the corresponding Chinese characters according to the extracted strokes. However, in order to eliminate interference during the writing process, an algorithm that can adaptive the writing trajectory according to the position of the writing board is required. Dynamic movement primitive (DMP) is an effective method for modelling robot movement behaviour and biological phenomena [24]. It has the advantage of being stable, simple and easy to generalise. It is possible to generalise a trajectory simply by modifying the position of the start and target points, and the shape of the generalised trajectory is same to the original demonstration. Because of its significant advantages, DMPs have been widely used and many studies have improved the learning and generalisation capabilities of DMPs [25]. In [26], Gaussian mixture model and Gaussian mixture regression were integrated to enhance DMPs' learning skills from multiple demonstrations. In [27], the learning skills are split into a series of sub-skills, thus the generalisation ability of the learned skills is improved. In [28], the framework of the DMPs is extended to accommodate real-time changes during task execution. In [29], a Riemannian-based DMP framework is proposed to learn and generalise multi-space data. In [30], a biomimetic controller is integrated with DMPs to facilitate the learning and adaptation of compliant skills. The authors of [31] proposed a new DMPs to solve the problems arising from the spatial scaling of DMPs in Cartesian space. The authors of [32] used Riemannian metrics to reformulate DMPs such that the resulting DMPs can be directly employed with quantities expressed as Symmetric Positive Definite (SPD) matrices. The study in [33] introduced a deep neural network in DMPs to address the invalidation of DMPs forcing terms. The authors of [34] developed a framework for the robot to learn both movement and muscle stiffness features. However, all these methods are difficult to adapt to the case where the start and target points are rotated simultaneously. The writing trajectory of the robot may be distorted when the writing board is rotated and flipped.

At the same time, the contact force between the pen tip and the writing board is an important piece of information during writing, especially when disturbed. Therefore, it is extremely necessary to design a writing system that combines stroke segmentation, trajectory generalisation, visual information and force information.

To address the above issues, the following work has been done. Firstly, inspired by the human mind-set of splitting strokes, a direction-based algorithm for extending and stitching stroke components was designed to extract the strokes of a Chinese character picture in a simple and fast way. Subsequently, some improvements were made to the DMPs. The rotation and translation matrices were fused into the framework of the DMPs so that the DMPs could accommodate the simultaneous rotation of the start and target points, i.e., it could accommodate the rotation and flipping of the writing board. Finally, an interference-resistant robotic Chinese character writing system was developed. The system uses segmented strokes as demonstrations and then combines the rotation and translation matrices calculated from visual information with improved DMPs to generalise the trajectories in real time. At the same



**Figure 1.** The connection and intersection of Chinese character “zhong”.



**Figure 2.** (a) The correct stroke division for the Chinese character “zhong”. (b) The incorrect stroke division after the improper handling of the connections.

time, the position is regulated online using admittance control based on force sensor information. Thus, the robot can write complete characters on the writing board despite a human’s random movement, rotation and flipping.

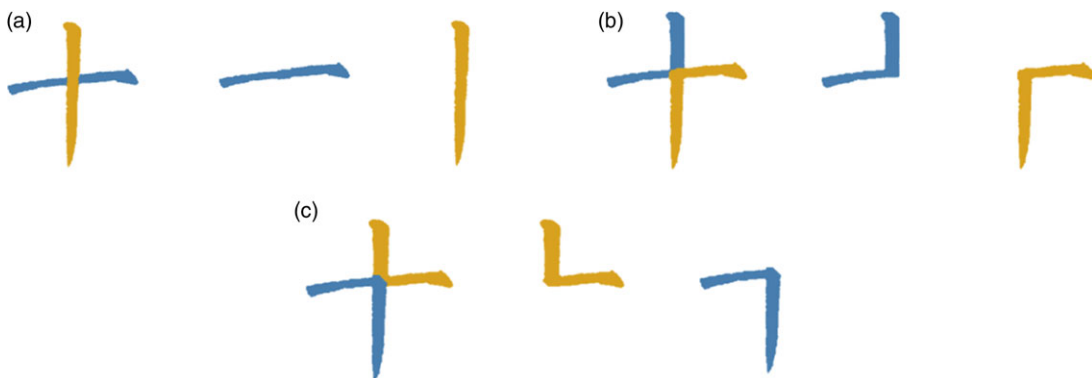
The rest of this article is presented below. Section 2 describes the proposed method in this article, including stroke division, force-position hybrid control and the designed system. The stroke division algorithm is divided into stroke component extraction and stroke component connection. Modified DMPs and admittance control are introduced in force-position hybrid control. Finally, a jam-resistant robotic writing system was designed based on the above algorithm. Section 3 provides sufficient experimental examples to demonstrate the effectiveness of the stroke division algorithm, modified DMPs, admittance control, and the designed system. Section 4 is the conclusion.

## 2. Proposed method

### 2.1. Stroke division

Stroke division is one of the difficulties in the study of Chinese characters because it is tough to learn the connections and intersections in Chinese characters. When the endpoints of two strokes are at the same point, as shown in Fig. 1, the point is defined as a connection. Improper recognising this point will tend to misidentify multiple strokes as the same stroke, as shown in Fig. 2. The intersection is the point formed when two strokes intersect, and improper handling of this point will result in incorrect recognition of the strokes, as shown in Fig. 3.

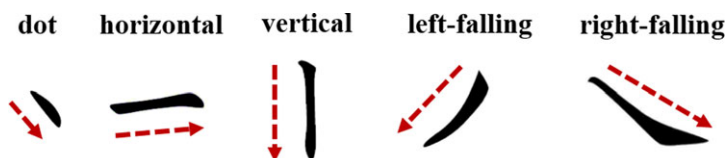
The Chinese character strokes are shown in Fig. 4. Analysing the strokes of Chinese characters, we can see that each stroke can be divided into one or more approximately straight curves. These curves can be summarised as dot, horizontal, vertical, left-falling and right-falling as shown in Fig. 5. These curves are named stroke components. Thus, each Chinese character stroke can be formed by combining one or more Chinese character stroke components. For example, as shown in Fig. 6, a Chinese character stroke “dot” can be seen as a combination of a Chinese character stroke component “dot”. A Chinese character stroke “vertical-turning” can be decomposed as a combination of a Chinese character stroke component “vertical” and a Chinese character stroke component “horizontal”. For another example, a Chinese character “horizontal-break-hook” can be seen as a combination of a “horizontal”, a “vertical” and a “dot”.



**Figure 3.** (a) The correct stroke division for the Chinese character “shi”. (b) and (c) The incorrect stroke division after the improper handling of the intersection.

丶	一	丨	ノ	㇏	㇏	㇏
丨	㇏	㇏	㇏	㇏	㇏	㇏
→	㇏	㇏	㇏	㇏	㇏	㇏
㇏	㇏	㇏	㇏	㇏	㇏	㇏

**Figure 4.** Chinese character strokes.



**Figure 5.** Chinese character stroke components.

As the Chinese character stroke components all extend in the same direction with no turning points, the Chinese character stroke components can be extracted when the direction of extension is known. At the same time, the determination of the direction of extension also deals with the issue of connections and intersections. A turning place connects two stroke components, and thus the turning identification is important in the basic component extraction. The intersection occurs when the strokes have been extended to different directions. When the different stroke components’ writing directions are determined, the stroke component extensions can cross the intersection in the specified direction to prevent misidentification at the intersection. Therefore, the stroke extraction algorithm has two parts: stroke components extraction and stroke components connection.

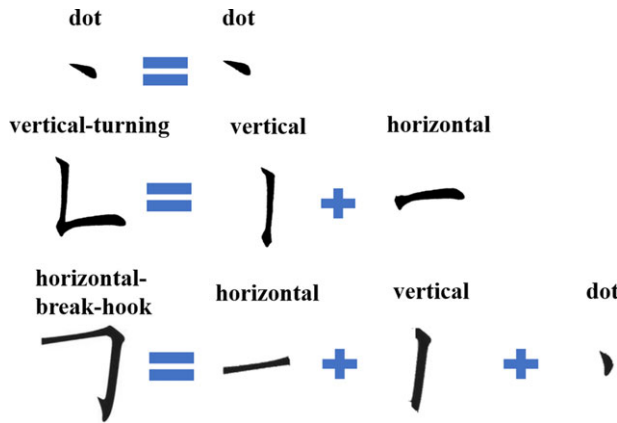


Figure 6. Chinese character stroke can be formed by combining one or more Chinese character stroke components.

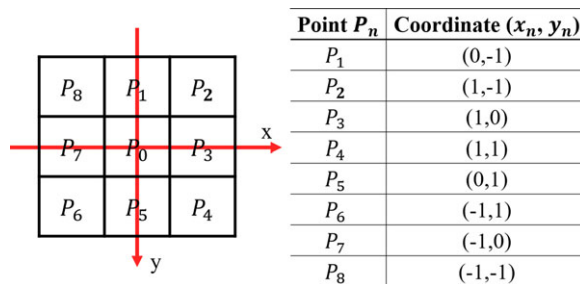


Figure 7. A set of eight neighbourhoods with  $P_0$  as the centroid.

2.1.1. Stroke components extraction

Firstly, the handwritten Chinese character images were binarised and refined. The algorithm for stroke components extraction is built on the basis of a refined image of a white Chinese character on a black background. The image is then traversed to find the white pixel, which is recorded as the starting point  $P_{sta}$  and treated as  $P_0$ . As shown in Fig. 7, for a set of eight neighbourhoods with  $P_0$  as the centroid, we set  $P_0$  as the origin and the length of the pixel as the unit length. The pixel points in this eight-neighbourhood are therefore named by the coordinates in which they are located. Starting from the pixel point above  $P_0$ , the pixels in a clockwise circle are each named  $P_n = (x_n, y_n)(n = 1, 2, 3, \dots, 8)$ . The process of extracting the strokes component is shown in Fig. 8. Define a direction vector  $\mathbf{D} = (\alpha, \beta)$  with an initial value of  $(0, 0)$ .  $\mathbf{D}$  describes the direction from the start pixel to the end pixel of the searched stroke component. The values of  $\mathbf{D}$  are normalised to obtain the coordinate points  $P_D$  in the eight-neighborhood coordinate system.

$$P_D = (x_D, y_D) \tag{1}$$

$$x_D = \frac{\alpha}{\sqrt{\alpha^2 + \beta^2}} \tag{2}$$

$$y_D = \frac{\beta}{\sqrt{\alpha^2 + \beta^2}} \tag{3}$$

The distance between  $P_D$  and the coordinates of each pixel in the eight neighbourhoods is

$$d_n = |P_n - P_D| \tag{4}$$

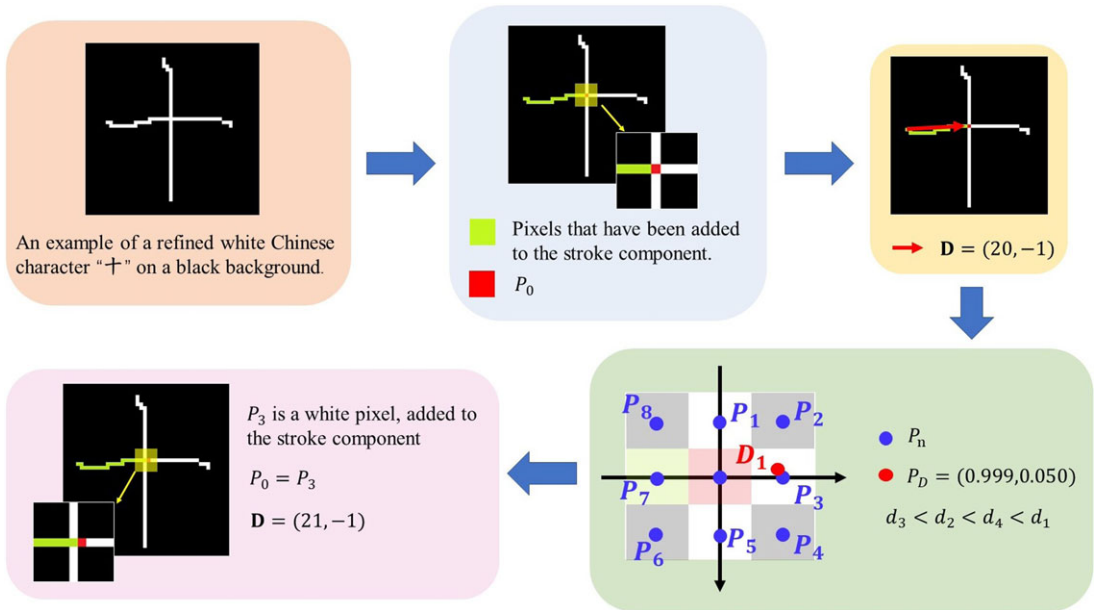


Figure 8. The process of extracting the strokes component.

Arrange  $d_1, d_2, \dots, d_n$  according to their values (from small to large values). To ensure that the stroke component extends in the direction indicated by  $\mathbf{D}$ , perform a sequential search for the  $P_n$  corresponding to the first 4  $d_n$ . If all the  $P_n$  picked out are black, the search stops. If  $P_n$  is a white pixel, extend the base stroke to  $P_n$  and update direction vector

$$\mathbf{D}_{\text{new}} = (\alpha_{\text{new}}, \beta_{\text{new}}) \tag{5}$$

$$\alpha_{\text{new}} = \alpha + x_n \tag{6}$$

$$\beta_{\text{new}} = \beta + y_n \tag{7}$$

Subsequently,  $P_n$  is used as the next  $P_0$  and  $\mathbf{D}_{\text{new}}$  as the next  $\mathbf{D}$  in order to continue extending the stroke component until the search stops.

However, the starting point  $P_{\text{sta}}$  is not always the endpoint of the stroke component. So after the extension along direction  $\mathbf{D}$  is complete, the direction vector  $\mathbf{D}$  needs to be reversed

$$\mathbf{D}_R = -\mathbf{D} \tag{8}$$

It is then extended to the opposite direction from the starting point  $P_{\text{sta}}$ , and finally, the two curves are joined to form a complete stroke component.

In order to prevent the same stroke component from being searched repeatedly, each pixel point is blacked out after it has been recorded as a point in the stroke component. However, this creates the interruption problem. As shown in Fig. 9, an intersection point is shared by multiple stroke components and when the intersection is blacked out, another stroke component passing through this intersection will be interrupted at the intersection. Therefore, we use two identical refined images  $I_1$  and  $I_2$  of white characters on a black background. Image  $I_1$  is blacked out at the recorded pixels and is used to find the starting point  $P_{\text{sta}}$ . This prevents repeated searching of the same basic stroke. Image  $I_2$  is not blacked out at the recorded pixels and is used to extend the stroke component so that the stroke components are not interrupted at the intersection point.

**Algorithm 1: Sorting of points**

**input:** stroke component  $SC$ ;  $SC$ 's starting point  $(x_{start}, y_{start})$ ;  $SC$ 's endpoint  $(x_{end}, y_{end})$ ;  
**output:** stroke component  $SC$

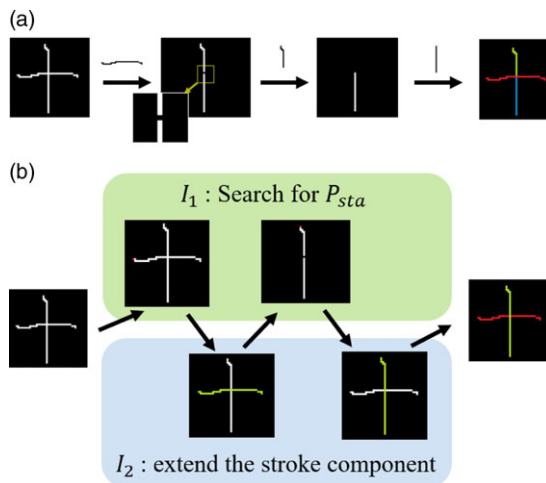
$x_{error} = x_{end} - x_{start}$ ;

$y_{error} = y_{end} - y_{start}$ ;

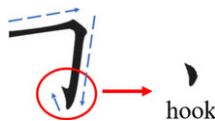
**if**  $(|x_{error}| > |y_{error}| \text{ and } x_{end} < x_{start})$  **or**  $(|x_{error}| < |y_{error}| \text{ and } y_{end} < y_{start})$  **then**

    | The points of  $SC$  are sorted in reverse order;

**end**



**Figure 9.** (a) The darkening at the intersection divides the vertical into two parts. (b) Using two identical images  $I_1$  and  $I_2$  can split the stroke components correctly.



**Figure 10.** The “hook” in Chinese character “horizontal-break-hook”.

**2.1.2. Stroke components connection**

When the end point of stroke component  $SC_1$  and the start point of stroke component  $SC_2$  are connected,  $SC_1$  and  $SC_2$  will make a stroke. Therefore, the starting and ending points of the stroke components need to be identified, i.e. the points between the connected stroke components. According to the conventional writing rules, the strokes are written from left to right or from top to bottom. The specific sorting algorithm is shown in Algorithm 1.

However, there is a special Chinese stroke called “hook” as shown in Fig. 10, which is written from bottom to top and from right to left. Similar to the stroke component “dot”, it is much shorter than the other basic strokes. It is also connected to the other stroke components. Thus, a stroke component is considered to be a “hook” if it satisfies the following conditions:

1. the end point is connected to the end points of the other stroke components;
2. the start point is not connected to the end points of any of the other stroke components;
3. the length is less than one-half the length of the average stroke component.

When combining strokes, the points of the “hook” need to be reverse ordered and then connected to other stroke components.

## 2.2. Force-position hybrid control

For a robotic writing system, the position control mode provides a limited performance, e.g. nonsmooth and nonsimilar to human’s writing characters. To write smoothly on the writing board, the contact force between the pen tip and the writing board needs to be controlled. Therefore, we use the force-position hybrid control. A modified DMP is used to control the position and posture of the robot’s end-effector. At the same time, the contact force between the pen tip and the writing board is controlled using admittance control. As a result, the robot can perform writing tasks.

### 2.2.1. Modified DMPs

Based on a spring damping model, the DMPs use a non-linear adjusting term to achieve the desired point attractor behaviour. It has the properties of a second-order dynamical system such as convergence at the target point, robustness to disturbances and generalisation in time and space. At the same time, the non-linear terms can be made to produce a continuous smooth trajectory. DMPs are classified as discrete DMPs and periodic DMPs. We use the discrete DMPs to model writing skills. The discrete DMPs model can be expressed as:

$$\tau \dot{v} = \alpha (y_g - y) - \beta v + f(x) \quad (9)$$

$$\tau \dot{y} = v \quad (10)$$

where  $\tau$  is the temporal scaling constant, and changing it allows the trajectory to generalise in the time dimension.  $\alpha$  and  $\beta$  represent the damping factor and the spring constant, respectively, and  $\alpha$  can generally be set to  $\alpha = \beta^2/4$ .  $y_g$  is the target position.  $y$  is the position of the demonstration.  $v$  is the velocity of the demonstration. The forcing term  $f(x)$  can be expressed as:

$$f(x, y_g) = \frac{\sum_{i=1}^N \psi_i \cdot \omega_i}{\sum_{i=1}^N \psi_i} \cdot x \cdot (y_g - y_0) \quad (11)$$

$$\psi_i = \exp(-h_i(x - c_i)^2) \quad (12)$$

$$\tau \dot{x} = -\alpha_x \cdot x \quad (13)$$

where  $\omega_i$  is the weight of the Gaussian kernel function,  $y_0$  is the starting point of the trajectory,  $c_i$  and  $h_i$  are the centre and width of the Gaussian kernel function, respectively,  $N$  denotes the number of Gaussian kernel functions, and  $\alpha_x$  is a positive gain coefficient.

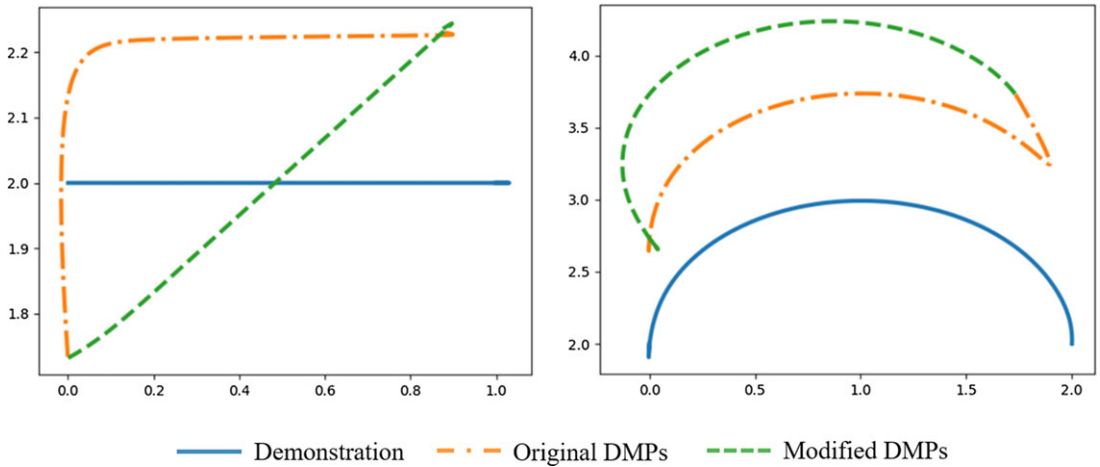
However, as shown in Fig. 11, when the start and target points are rotated, the original DMP cannot rotate the generalised trajectory in its original shape and the trajectory is distorted. In order to apply the DMPs to this case where both the start and target points are rotated, we have improved the original DMPs inspired by [31]. The generalised space is obtained by rotating the original space. Once the rotation matrix between the two spaces is known, the corresponding rotation can be applied to the non-linear term. Thus, equation (9) becomes

$$\tau \ddot{y} = \alpha_y (\beta_y (y_{g,1} - y) - \dot{y}) + s \cdot \mathbf{R} \cdot f \quad (14)$$

$$y_{g,1} = \mathbf{R} \cdot y_{g,0} + \mathbf{t} \quad (15)$$

$$y_{0,1} = \mathbf{R} \cdot y_{0,0} + \mathbf{t} \quad (16)$$





**Figure 11.** Comparison of original DMPs and modified DMPs. For linear and curvilinear demonstrations, generalisations are carried out using the original DMPs and modified DMPs, respectively. The trajectories generalised from the original DMPs have distortions. The trajectories generalised by the improved DMPs conform to the expected trajectories.

$$s = \frac{\|y_{g,1} - y_{0,1}\|}{\|y_{g,0} - y_{0,0}\|} \tag{17}$$

where  $y_{g,1}$  and  $y_{g,0}$  are the target points in the generalised space and the original space, respectively.  $y_{0,1}$  and  $y_{0,0}$  are the starting points in the generalised space and the original space, respectively.  $s$  is the scaling constant.  $\mathbf{R}$  is the rotation matrix and  $\mathbf{t}$  is the translation matrix. Throughout this article,  $\mathbf{R}$  and  $\mathbf{t}$  are calculated from visual information.

### 2.2.2. Admittance control

The admittance model is defined as follows:

$$M_A \ddot{x}_A + D_A \dot{x}_A + K_A(x_A - x_0) = f_h \tag{18}$$

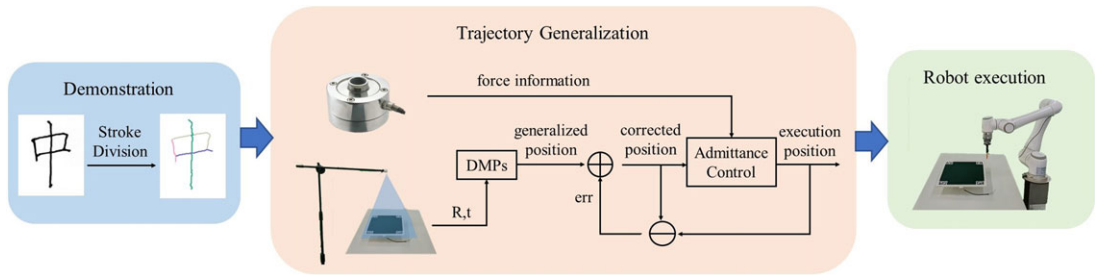
where  $M_A$  is the quality matrix,  $D_A$  is the damping matrix and  $K_A$  is the spring factor matrix.  $f_h$  is the traction of the demonstrator.  $x_A$  is the pose of the robot end-effector,  $\dot{x}_A$  is the speed and  $\ddot{x}_A$  is the acceleration.  $x_0$  is the original point of the pose. The robot end-effector needs to move on the writing board during the writing process, so  $x_A = x_0$ . The admittance model can be simplified as:

$$M_A \ddot{x}_A + D_A \dot{x}_A = f_h. \tag{19}$$

Therefore, the acceleration can be calculated using the traction force  $f_h$  and then the velocity and position can be calculated by integration to obtain the trajectory. In this article,  $f_h$  is the contact force between the pen tip and the writing board, with an expected value of 2N.

### 2.3. Designed system

Based on the above approach, we have designed a multi-sensor-based robotic writing system. As shown in Fig. 12, at the demonstration stage, the strokes are extracted from the demonstrations. During the trajectory generalisation phase, the camera detects changes in the pose of the writing board and calculates the rotation matrix  $\mathbf{R}$  and the translation matrix  $\mathbf{t}$ . The DMPs model calculates the generalised trajectory points based on the  $\mathbf{R}$  and  $\mathbf{t}$  and feeds the trajectory points into the



**Figure 12.** Robotic writing system.

admittance control model. The admittance control algorithm calculates the execution position based on the given trajectory points and real-time force information and outputs the execution position to the robot. At the same time, the execution position will be different from the desired position and in the next step this differential will be used to correct the trajectory points in advance to improve accuracy.

### 3. Experiment

#### 3.1. Stroke extraction

In order to verify the accuracy of Chinese character stroke extraction, we randomly selected 100 hand-writing Chinese characters with different structures and complexities for experiments. As shown in Fig. 13, the experiments show that the algorithm works well. Statistically, 93% of the strokes of hand-written Chinese characters can be extracted accurately, with an average speed of 0.4 s per image. The characters boxed in Fig. 13 are characters with inaccurate stroke recognition. After studying them, we have divided them into three categories. Category 1 is for those boxed in red, whose strokes are broken. This means that a single stroke has been split into multiple strokes or is incompletely recognised. Those boxed in green are in category 2: misconnected strokes. This means that a stroke has been incorrectly connected to another stroke. Boxed in blue is category 3, there are redundant strokes. Excess strokes take the form of burrs consisting of just one or two pixels. Analysis of these three categories shows that almost all recognition errors happen in the refinement process. When inaccurate refinement leads to a change in the direction of the strokes, deformations are observed at the intersection and junction points of the Chinese character's skeleton, errors in category 1 are likely to occur. During the writing process, variations in pen pressure or ink leakage can result in changes in the thickness of strokes, leading to the occurrence of jaggies in the refinement process, errors in category 3 are likely to occur. Errors in category 2 are caused by inaccurate refinement and the poor writing where multiple interconnected or intersecting stroke components share a consistent direction.

#### 3.2. Modified DMPs

There are two methods to generalise a trajectory based on disturbances in real time. Method 1 firstly generalises the new trajectory using the original DMPs and then adjusts the new trajectory using rotation and translation matrices,  $\mathbf{R}$  and  $\mathbf{t}$ . Method 2 uses the modified DMPs to generalise the trajectory directly based on  $\mathbf{R}$  and  $\mathbf{t}$ . We set up a writing track and use visual information to obtain the pose of the writing board to calculate  $\mathbf{R}$  and  $\mathbf{t}$ . These two methods are then used to make the trajectory follow the desired pose on the writing board. The effects of the two methods are shown in Fig. 14. It can be seen that the trajectory output of method 1 is more oscillating, not smooth enough and has large errors. Method 2 outputs a smoother trajectory with less oscillation, which is more effective. In comparison to Method 1,

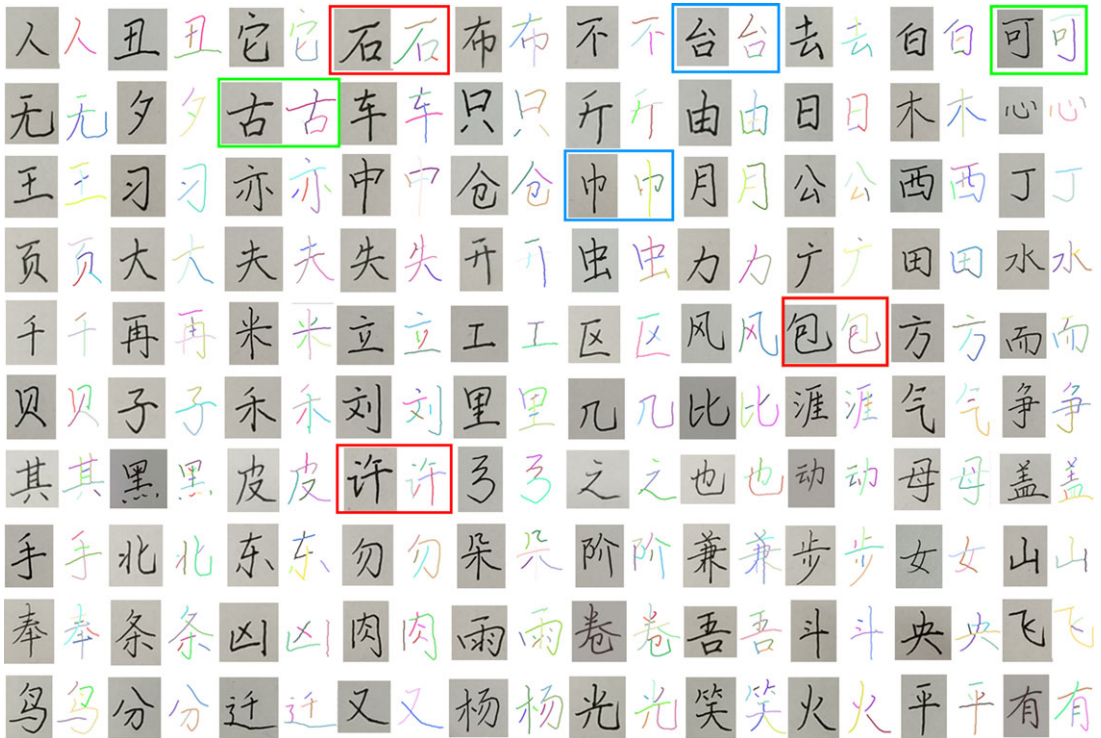
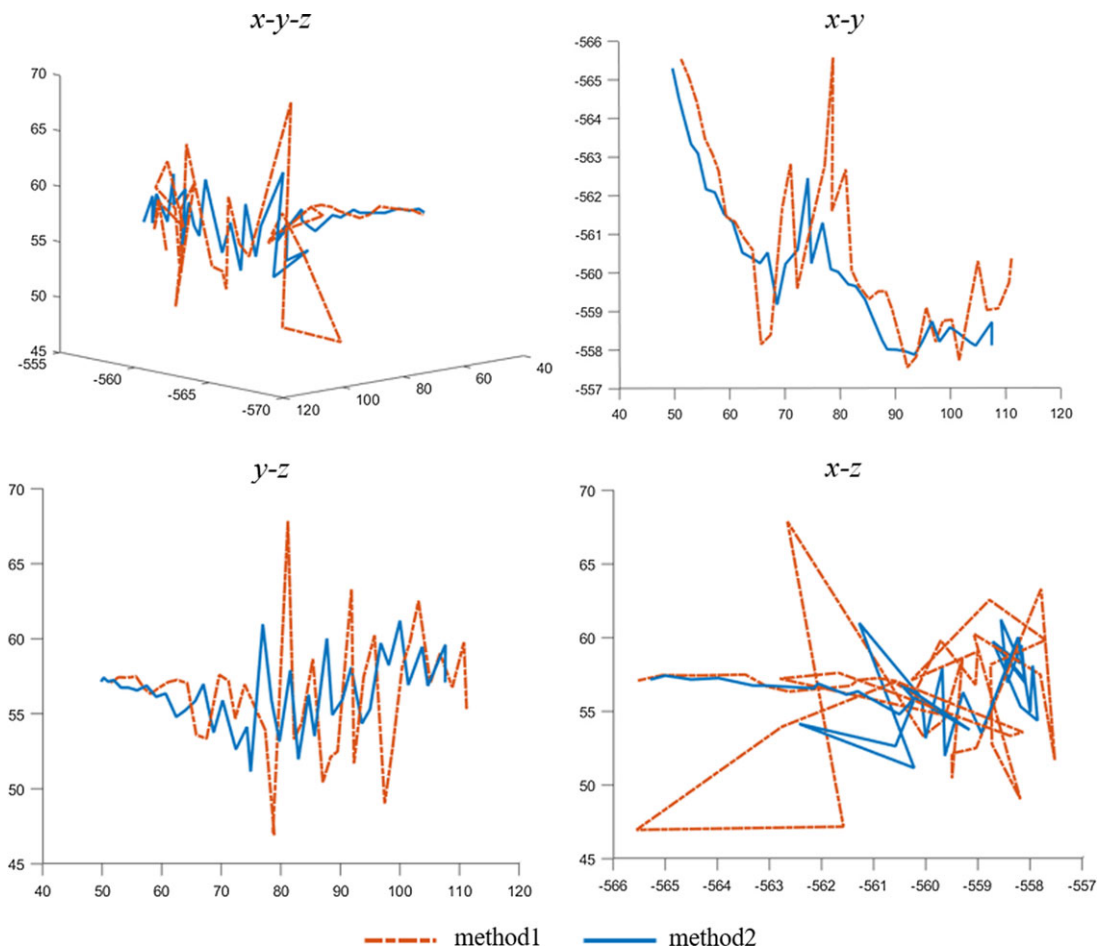


Figure 13. Stroke extraction results. Different colours indicate different strokes.

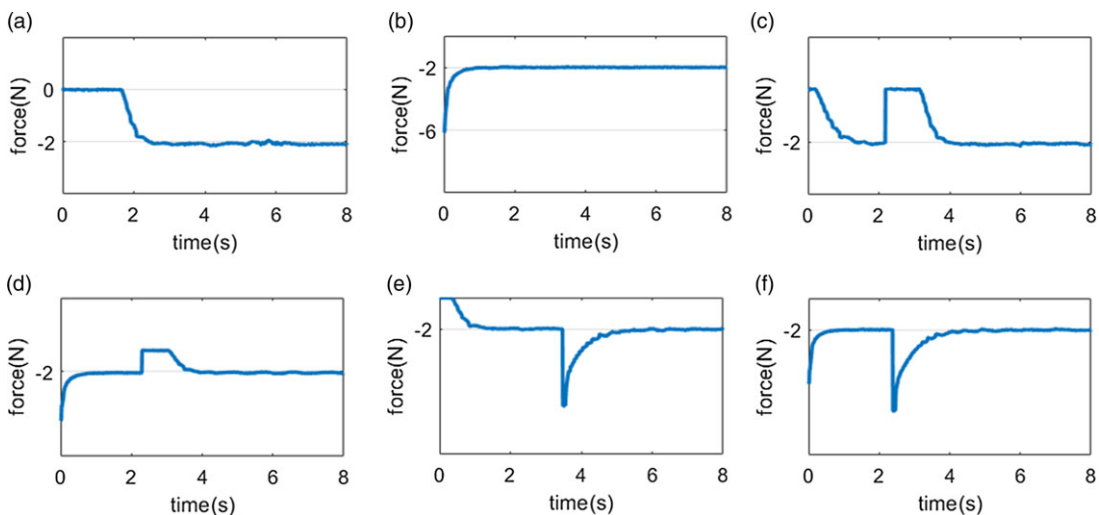
the trajectories produced by Method 2 exhibit a difference of up to 7.1977 mm in the z-direction, which poses a potential safety hazard in practical applications, as it could result in the inadvertent puncturing of the writing surface. We analysed the reasons for the large differences in the results between the two methods. When using method 1, outputs of the original DMP have errors in practice, such as distortion of the trajectory and robot movement errors. Also,  $\mathbf{R}$  and  $\mathbf{t}$  are calculated from visual information and have errors. Because  $\mathbf{R}$  and  $\mathbf{t}$  do not fall within the framework of DMPs, the multiplication with the trajectory points is not constrained by DMPs, leaving the trajectory with a large offset. Whereas method 2 incorporates  $\mathbf{R}$  and  $\mathbf{t}$  into the framework of DMPs, the generalisation of DMPs will change with  $\mathbf{R}$  and  $\mathbf{t}$ . At the same time, the trajectory does not produce large offsets due to the constraints of the DMPs framework.

### 3.3. Admittance control

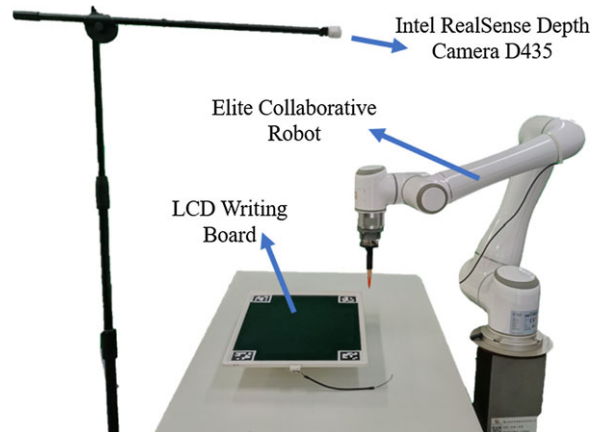
We have measured the contact force between the pen tip and the writing board when a person writes. We find that the best writing performance is achieved when the force sensor has  $2N$  contact force on the z-axis. If the contact force is not  $2N$  when the robot arm moves to the specified position, the admittance control algorithm can adjust the end position of the robot arm in time to guarantee the contact force always be  $2N$ . As shown in Fig. 15, when the initial contact force is 0, i.e. when the end-effector is not touching the writing board, the admittance control moves the end-effector closer to the writing board until the contact force converges to  $2N$ . When the initial contact force is  $6N$ , the admittance control policy steers the end-effector moving away from the writing board until the contact force is stabilised at  $2N$ . The admittance control also adjusts the contact force in time when there is any interference during the adjustment process.



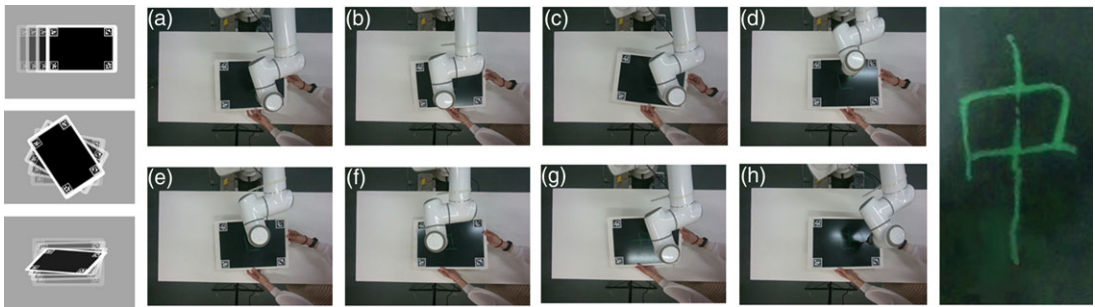
**Figure 14.** The effect of the two methods.



**Figure 15.** The effect of the admittance control.



**Figure 16.** *Experimental platform.*



**Figure 17.** *The results of the experiment.*

### 3.4. Application

In this experiment, we verify the effectiveness of the designed interference-resistant robotic Chinese character writing system. A human randomly writes a Chinese character. Then the robot imitates the human handwriting on the writing board. While the robot writes, the human randomly moves, turns and flips the board, and the robot can still follow the posture of the board to adjust its trajectory. As shown in Fig. 16, the experimental platform is composed of an Elite Collaborative Robot EC66, an Intel RealSense Depth Camera D435 and a LCD Writing Board. The Elite Collaborative Robot EC66 is a lightweight and flexible collaborative robot. Its weight is 17.5 kg and the largest payload is 6 kg. It runs efficiently and can reach a maximum speed of 2.8 m/s, with a soft and consistent trajectory. As a collaborative robot, it enables safe human–robot interaction with a high degree of reliability and safety. It can therefore complete different tasks while ensuring the safety of humans. With a range of 10 m and a wide field of view, the Intel RealSense Depth Camera D435 has a wide range of applications in robotics development. By affixing four calibration papers to the four corners of the writing board, it is able to estimate the posture of the writing board very well in this experiment, i.e. to recognise human interference. During the robot's writing process, the human randomly and continuously moves, rotates and flips the writing board. The robot can write complete Chinese characters on the writing board.

The results of the experiment are shown in Fig. 17. The robot is able to adjust its end trajectory in time to write complete and accurate Chinese characters on the writing pad under human interferences.

#### 4. Conclusion

In this article, we designed a robotic writing system that allows a robot to imitate human handwriting even with unexpected interruptions. We first designed a new stroke extraction algorithm that extracts stroke components based on the direction of extension, and then connected the stroke components according to the writing direction of the stroke. To cope with random interference, the original DMPs model was improved such that the generalisation function could be adapted to the situations when the writing board was rotated and flipped. The writing system was then designed by combining visual information, admittance control and an improved DMP to enable the robot to accurately imitate human handwriting in the face of interference.

In this article, we have presented an interference-resistant robotic writing system that can also be used in applications such as grinding and cleaning of complex recesses in workpieces. In the future, we will improve the current visual measurement module to make this system more accurate and better applicable.

**Author contributions.** Xian Li, Chenguang Yang, Sheng Xu and Yongsheng Ou proposed the research project; Xian Li conducted the review and wrote the first draft; Xian Li, Chenguang Yang, Sheng Xu and Yongsheng Ou revised the manuscript; and Chenguang Yang, Sheng Xu and Yongsheng Ou provided advice and supervision.

**Financial support.** This work was supported in part by National Nature Science Foundation of China (NSFC) under Grant U20A20200 and Major Research Grant No. 92148204, in part by Guangdong Basic and Applied Basic Research Foundation under Grants 2019B1515120076 and 2020B1515120054, in part by Industrial Key Technologies R & D Program of Foshan under Grant 2020001006308 and Grant 2020001006496.

**Competing interests.** The authors declare no competing interests exist.

**Ethical approval.** Not applicable.

#### References

- [1] Z. Li, S. Li and X. Luo, "An overview of calibration technology of industrial robots," *IEEE/CAA J. Autom. Sin.* **8**(1), 23–36 (2021).
- [2] H. Yin, P. Alves-Oliveira, F. S. Melo, A. Billard and A. Paiva, "Synthesizing Robotic Handwriting Motion by Learning from Human Demonstrations." *Proceedings of the 25th International Joint Conference on Artificial Intelligence, No. CONF* (2016).
- [3] H. Zeng, Y. Huang, F. Chao and C. Zhou, "Survey of robotic calligraphy research," *CAAI Trans. Intell. Syst.* **11**(1), 15–26 (2016).
- [4] F. Chao, Y. Huang, X. Zhang, C. Shang, L. Yang, C. Zhou, H. Hu and C.-M. Lin, "A robot calligraphy system: From simple to complex writing by human gestures," *Eng. Appl. Artif. Intel.* **59**, 1–14 (2017).
- [5] S. Wang, J. Chen, X. Deng, S. Hutchinson and F. Dellaert, "Robot Calligraphy Using Pseudospectral Optimal Control in Conjunction with a Novel Dynamic Brush Model." *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE (2020) pp. 6696–6703.
- [6] G. Wang, Z. Liao and X. Chen, "Robot manipulator lettering technology and motion," *J Chongq. Univ.* **26**(12), 6–9 (2003).
- [7] X. Gong and J. Wang, "Drawing Chinese character based on continuous path control of IRB 140 industrial robot," *Modern Manuf. Eng.* (2), 135–137(2010).
- [8] Z. Chuanjian, L. Chunmei and H. Jun, "Functional design and realization of writing for motoman-up6 robots," *Mach. Buildi. Autom.* **40**(2), 119–121 (2011).
- [9] Y. Yang, W. Chen, L. Zhou, B. Zheng, W. Xiao, Y. Huang and Z. Sun, "A Hybrid Control Framework Teaching Robot to Write Chinese Characters: From Image to Handwriting," *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, IEEE (2021) pp. 1161–1166.
- [10] F. Yao and G. Shao, "Modeling of Ancient-style Chinese Character and Its Application to CCC Robot," *2006 IEEE International Conference on Networking, Sensing and Control.*, IEEE (2006) pp. 72–77.
- [11] F. Yao, G. Shao and J. Yi, "Extracting the trajectory of writing brush in Chinese character calligraphy," *Eng. Appl. Artif. Intel.* **17**(6), 631–644 (2004).
- [12] J. H. Lam and Y. Yam, "Application of Brush Footprint Geometric Model for Realization of Robotic Chinese Calligraphy," *2011 2nd International Conference on Cognitive Infocommunications (CogInfoCom)*, IEEE (2011) pp. 1–5.

- [13] J. H. Lam and Y. Yam, “Stroke Trajectory Generation Experiment for a Robotic Chinese Calligrapher Using a Geometric Brush Footprint Model,” *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2009) pp. 2315–2320.
- [14] J. Li, W. Sun, M. Zhou and X. Dai, “Teaching a Calligraphy Robot Via a Touch Screen,” *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE (2014) pp. 221–226.
- [15] F. Lin and X. Tang, “Dynamic Stroke Information Analysis for Video-based Handwritten Chinese Character Recognition,” *Proceedings Ninth IEEE International Conference on Computer Vision*, IEEE (2003) pp. 695–700.
- [16] Y. Sun, H. Qian and Y. Xu, “Robot Learns Chinese Calligraphy from Demonstrations,” *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE (2014) pp. 4408–4413.
- [17] F. Chao, Y. Huang, C.-M. Lin, L. Yang, H. Hu and C. Zhou, “Use of automatic Chinese character decomposition and human gestures for Chinese calligraphy robots,” *IEEE Trans. Hum. Mach. Syst.* **49**(1), 47–58 (2018).
- [18] F. Cao, Z. Wu, X. Ao and M. Zhou, “Vectorization of Qi Gong calligraphy,” *J. Chin. Inf. Process.* **24**(6), 97–102 (2010).
- [19] X. Wang, X. Liang, L. Sun and M. Liu, “Triangular Mesh Based Stroke Segmentation for Chinese Calligraphy,” *2013 12th International Conference on Document Analysis and Recognition*, IEEE (2013) pp. 1155–1159.
- [20] L. Gan, W. Fang, F. Chao, C. Zhou, L. Yang, C.-M. Lin and C. Shang, “Towards a Robotic Chinese Calligraphy Writing Framework,” *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE (2018) pp. 493–498.
- [21] T.-Q. Wang, X. Jiang and C.-L. Liu, “Query pixel guided stroke extraction with model-based matching for offline handwritten Chinese characters,” *Pattern Recogn.* **123**, 108416 (2022).
- [22] Z. Ju, X. Ji, J. Li and H. Liu, “An integrative framework of human hand gesture segmentation for human–robot interaction,” *IEEE Syst. J.* **11**(3), 1326–1336 (2015).
- [23] C.-C. Chang, “A library for support vector machines. 2001. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [24] A. Ijspeert, J. Nakanishi and S. Schaal, “Learning Attractor Landscapes for Learning Motor Primitives,” *In: Advances in Neural Information Processing Systems*. vol. 15 (2002).
- [25] L. Koutras and Z. Doulgeri, “Dynamic Movement Primitives for Moving Goals with Temporal Scaling Adaptation,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2020) pp. 144–150.
- [26] C. Yang, C. Chen, W. He, R. Cui and Z. Li, “Robot learning system based on adaptive neural control and dynamic movement primitives,” *IEEE Trans. Neur. Netw. Learn. Syst.* **30**(3), 777–787 (2018).
- [27] C. Yang, C. Zeng, Y. Cong, N. Wang and M. Wang, “A learning framework of adaptive manipulative skills from human to robot,” *IEEE Trans. Ind. Inform.* **15**(2), 1153–1161 (2018).
- [28] A. S. Anand, A. Østvik, E. I. Grøtli, M. Vagia and J. T. Gravdahl, “Real-time temporal adaptation of dynamic movement primitives for moving targets,” *2021 20th International Conference on Advanced Robotics (ICAR)*, IEEE (2021) pp. 261–268.
- [29] Z. Liao, G. Jiang, F. Zhao, Y. Wu, Y. Yue and X. Mei, “Dynamic skill learning from human demonstration based on the human arm stiffness estimation model and Riemannian DMP,” *IEEE/ASME Trans. Mechatron.* **28**(2), 1149–1160 (2022).
- [30] C. Zeng, X. Chen, N. Wang and C. Yang, “Learning compliant robotic movements based on biomimetic motor adaptation,” *Robot. Auton. Syst.* **135**, 103668 (2021).
- [31] L. Koutras and Z. Doulgeri, “A Novel DMP Formulation for Global and Frame Independent Spatial Scaling in the Task Space,” *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, (2020) pp. 727–732.
- [32] F. J. Abu-Dakka and V. Kyrki, “Geometry-aware Dynamic Movement Primitives,” *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE (2020) pp. 4421–4426.
- [33] T. Wang, L. Yan, G. Wang, X. Gao, N. Du and I.-M. Chen, “Learning from Demonstration Using Improved Dynamic Movement Primitives,” *2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA)*, (2021) pp. 2130–2135.
- [34] C. Yang, C. Zeng, C. Fang, W. He and Z. Li, “A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills,” *IEEE/ASME Trans. Mechatron.* **23**(3), 1193–1203 (2018).