# A Robot Calligraphy System: From Simple to Complex Writing by Human Gestures

Chao, Fei; Huang, Yuxuan; Zhang, Xin ; Shang, Changjing; Yang, Longzhi; Zhou, Changle ; Hu, Huosheng ; Lin, Chih-Min

*Published in:*
Engineering Applications of Artificial Intelligence

*DOI:*
10.1016/j.engappai.2016.12.006

*Publication date:*
2017

*Citation for published version (APA):*
Chao, F., Huang, Y., Zhang, X., Shang, C., Yang, L., Zhou, C., Hu, H., & Lin, C.-M. (2017). A Robot Calligraphy System: From Simple to Complex Writing by Human Gestures. *Engineering Applications of Artificial Intelligence*, *59*, 1-14. https://doi.org/10.1016/j.engappai.2016.12.006

# A Robot Calligraphy System: From Simple to Complex Writing by Human Gestures

Fei Chao[a], Yuxuan Huang[a], Xin Zhang[a], Changjing Shang[b], Longzhi Yang[c], Changle Zhou[a], Huosheng Hu[a,d], Chih-Min Lin[a,e]

[a] *Cognitive Science Department, Fujian Provincial Key Laboratory of Machine Intelligent and Robotics, School of Information Science and Engineering, Xiamen University, 361005, China. Corresponding author: Fei Chao. Email: fchao@xmu.edu.cn. Tel: +86 592 2580192*
[b] *Department of Computer Science, Aberystwyth University, UK*
[c] *Computer Science and Digital Technologies Department, Northumbria University, UK*
[d] *School of Computer Science and Electronic Engineering, University of Essex, UK*
[e] *Department of Electrical Engineering, Yuan Ze University, Taiwan*

---

**Abstract**

Robotic writing is a very challenging task and involves complicated kinematic control algorithms and image processing work. This paper, alternatively, proposes a robot calligraphy system that firstly applies human arm gestures to establish a font database of Chinese character elementary strokes and English letters, then uses the created database and human gestures to write Chinese characters and English words. A three-dimensional motion sensing input device is deployed to capture the human arm trajectories, which are used to build the font database and to train a classifier ensemble. 26 types of human gesture are used for writing English letters, and 5 types of gesture are used to generate 5 elementary strokes for writing Chinese characters. By using the font database, the robot calligraphy system acquires a basic writing ability to write simple strokes and letters. Then, the robot can develop to write complex Chinese characters and English words by following human body movements. The classifier ensemble, which is used to identify each gesture, is implemented through using feature selection techniques and the harmony search algorithm, thereby achieving better classification performance. The experimental evaluations are carried out to demonstrate the feasibility and performance of the proposed method. By following the motion trajectories of the human right arm, the end-effector of the robot can successfully write the English words or Chinese characters that

correspond to the arm trajectories.

*Keywords:* Robotic writing, Robotic Calligraphy, Human-robot interaction, Human gesture recognition, Classifier ensemble

*2010 MSC:* 00-01, 99-00

## 1. Introduction

Handwriting is a highly demanding task involving both dynamics and kinematics, and is therefore normally regarded as a typically human motion [1, 2]. The kernel technology of robotic writing is to combine a number of basic actions (for instance, letters of the alphabet or Chinese character strokes) in order to generate complex functions (such as full sentences). A secondary goal would be to optimise the visual quality of the robotic output, with the help of human expert selection. Moreover the process should be easily adaptable, that is human users require robots to be able to quickly learn to handle new characters, or new motions. Such kernel technology can be exported to many other domains, particularly where there is a high demand for robotic imitation of repeated human movements. Notable examples would include medical rehabilitation training, where robotic movements could take advantage of scaling and repeated movements to assist patient to rehabilitate from small to large movements, as well as customised painting in automotive design finishes, or industrial welding of non-linear specialised shapes. With this in mind, the following work on robotic writing should be viewed as the test bed for a much wider range of practical applications.

The process of robotic handwriting requires the robot to obtain trajectory information, whether the strokes of Chinese characters, or the shape of English letters. A number of recent approaches have applied direct programming methods to embed a font database within the robot's control systems, which require complicated mathematical calculations and image processing work [3, 4]. However, the imitation of human actions is considered as an effective learning method to transfer skills and knowledge from human beings to robots [5, 6, 7, 8, 9].

A wide range of applications using 3D human activity recognition has been introduced in recent years [10]. It is very useful for robots to acquire new skills without the need of complex programming and implementations [11]. Human users also prefer a convenient and natural way to directly control robots to copy characters and letters [12, 13, 14]. In particular, applying pen-tablets to obtain trajectory information [15] is also a fast and even more natural way to write [16, 17, 18]. However, using human gestures will provide more fine information towards controlling robots to write characters, e.g. wrist and elbow positioning data can be used to support robotic posture control. Beyond robotic writing, the ability to learn general human gestures should allow the proposed method to be adapted to handle dynamic robotic manipulations, such as the capturing and grasping of physical objects. Further, human gesture information can be effectually represented by medical EEG or EMG signals. Thus, this research grounds for EEG or EMG controlled robotic writing, a practical version of mind control.

Strokes of both Chinese characters and English letters can be represented by human arm trajectories. English consists of 26 different letters, which can be represented by 26 classes of human arm gestures. All Chinese characters are constructed by strokes. Yao et al. used 28 strokes to construct all of the Chinese characters [19]. Compared with writing English letters by hand to writing Chinese characters by hand, robots not only need to know how to write the strokes of a character, but also need to consider the layout of each character's stroke. In addition, if human gestures are applied to represent strokes and letters, the gesture recognition problem must still be considered. In particular, gesture information is presented by a large group of three-dimensional points; thus, a recognition mechanism is required to identify each gesture precisely. The advantage of human gestures is to build a robotic action database through imitations. A robot develops a more complex action by assembling simple actions from the database. Thus the more actions there are included in the database, the better the resulting operational ability the robot can achieve.

This paper proposes a novel approach to robotic handwriting based on our initial preliminary work [20, 14]. In the previous work, all the strokes had to be pre-programmed by human engineers, and only the writing actions were controlled through direct human gestures. The previous work had the obvious limitation that significant additional human programming was required to simply add a new font to the robot's database.

The work reported in [14] successfully supported free writing without repeated training or complex programming. However, because the robotic arm simply followed the demonstrator's movements, it was very difficult to improve the writing quality of the strokes. In this paper, the trajectories of human hand movements have two uses: (1) the trajectories can be recognized by a robot's classification methods; and (2) the trajectories are the font shapes of the characters themselves. A number of different methods to classify gestural expressions have been reported in the literature [21, 22, 23, 24, 25, 26], including "human gesture corpora based methods" [27], "Dynamic Bayesian Networks" [28], "Gaussian mixture modeling" [29], "3D extremity movement observation" [30], and "Hidden Markov Models" [31][32]. However, this research is inspired from Schumacher et al.'s work [33], i.e. the problem is addressed by classifying trajectory segments comprising a fixed number of sampling points of human gestures.

Generally speaking, any conventional classifier could be used to recognize human hand gestures. A classifier ensemble can improve the performance of a single classifier system. However, an ensemble with too many classifiers may demand a large computational time. Classifier Ensemble Reduction (CER) aims to reduce the redundancy in a pre-constructed classifier ensemble, so as to form a much reduced subset of classifiers that can still deliver the same classification results [21, 34, 35]. It is an intermediate step between ensemble construction and decision aggregation. Efficiency is one of the obvious gains from CER.

Removing redundant ensemble members may also lead to improved diversity within the group, and further increase the prediction efficiency of the ensemble [36]. Existing literature approaches include techniques that employ clustering [37] to discover groups of models that share similar predictions, and subsequently prune each cluster separately. Other approaches use “Reinforcement Learning” [38] and “Multi-label Learning” [39] to remove redundancy. In this paper, a new approach for CER that builds upon the ideas from existing feature selection techniques [34, 40] is applied to classify a human demonstrator’s gestures, so as to achieve a higher recognition rate for robotic writing.

In this work, a three-dimensional vision sensor, “Kinect”, is deployed to detect human right hand gestures. Kinect devices are widely applied in many robotic systems [41, 42]. The human hand’s trajectories must be consistent with the character’s trajectories. The robot system captures the human gestures, and controls the robotic arm to write the designated trajectories. In particular, the captured trajectories are then converted to an array of hand trajectory data. A novel reduced classifier ensemble for recognition is used to improve the gesture recognition accuracy. The classifier ensembles are known to usually improve recognition performance in a wide range of pattern recognition tasks [34]. A robot with a five DOFs arm receives the captured stroke trajectories, and kinematic algorithms are used to convert the stroke trajectories to the arm’s joint values; then, the robot completes the writing task. This approach reduces the complexity of creating robotic writing, thereby enabling robots to exhibit higher flexibility. Additionally, the robotic writing guided by human gestures can introduce a natural and convenient way to control robots to execute complicated tasks. The main contributions of this paper are summarized as follows:

- The method for automated generation of robot’s font database of Chinese character strokes and English letters from human arm’s gestures (Sections 2 and 3.2).
- The method for empowering the robot’s Chinese character and English word writing ability through the exploitation of the learned font database and given human gestures (Section 3).

The remainder of this paper is organised as follows. Section 2 describes the proposed framework and methodology used for robotic hand writing. In

Section 3, human hand gestures and robot arm control are introduced. Section 4 presents the experimental results and discusses their implications. Finally, a brief conclusion and potential future work are given in Section 5.

## 2. The Proposed Approach

### *2.1. Robotic Handwriting Framework*

Fig. 1 describes the framework of the robotic handwriting. First of all, a human demonstrator stands in front of the robotic system. The human uses one arm to perform predefined poses. The Gesture Sampling module is implemented by a Kinect device, which only captures the skeleton information of the human's poses. The skeleton information is sent to the Trajectory Capture module, in which the captured gestures are presented in 2-dimensional point arrays of the human's right arm trajectories. Then, the remaining approaches are divided into two phases: (1) the training phase, which includes classifier learning, classifier ensemble reducing, and obtaining trajectory information; and (2) the control phase, which uses the reduced classifier ensemble to identify the human gestures, and invokes the obtained trajectory information to write the identified strokes and letters.

In the first phase, the trajectory's point arrays are retained in a Training Dataset module. The dataset is then applied to train a classifier ensemble. A new ensemble reduction method, rather than a conventional ensemble approach, is applied to train the classifier ensemble. The new method is implemented by feature selection and harmony search techniques [34]. After the training, a reduced size classifier ensemble with high recognition accuracy is obtained for the second phase. Note that to gather enough training data, the demonstrator must repeat the gesturing of each letter and Chinese stroke many times. Additionally, the captured gestures are also retained in a Font Database module, from which the robot can obtain trajectory information. Therefore when the classifier ensemble produces a prediction of an input gesture, the robot can search the database for the font information that corresponds to the prediction. In

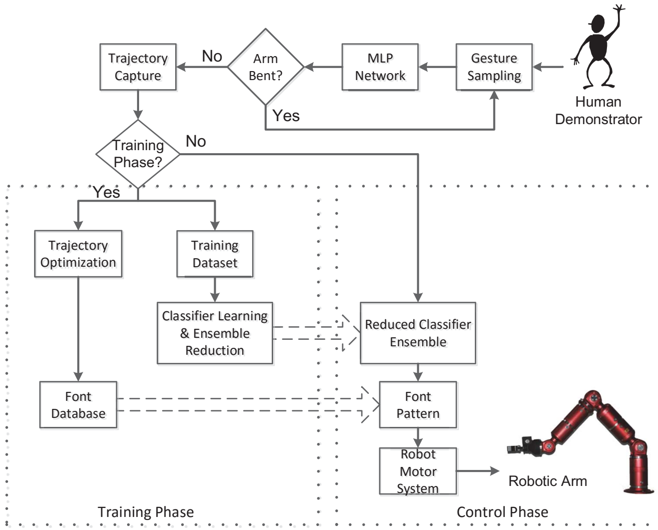


Figure 1: The flowchart of the robotic handwriting. The approach consists of the training phase and the control phase.

order to beautify the font, a Trajectory Optimization module is applied to filter unexpected noise.

In the second phase, the human demonstrator does not need to repeat the gesturing of predefined patterns. The demonstrator merely performs each stroke and letter in sequence. The Kinect device converts the gesture trajectories to skeleton data that are directly sent to the reduced classifier ensemble. Because the classifier ensemble has been trained in the first phase, the ensemble can generate the type of stroke or letter that is related to the gesture. Then, the Font Pattern module finds the saved trajectories from the first phase. The robot motor system applies inverse kinematic calculations to convert the trajectories to the robotic arm's joint values. The robot finally executes the arm joint values to write the strokes and letters.

Two different behaviorial patterns are applied to control the robot. (1) For writing Chinese characters: Chinese characters are disassembled into five elementary strokes. In this case, each Chinese character's writing is formed by writing different strokes in different positions. Thus, five classes of human gestures are assigned to five types of strokes. In addition, each gesture's starting and end positions are also kept. To implement the entire character, a human demonstrator performs diverse poses to represent the corresponded strokes. (2) For writing English letters: the demonstrator merely performs the English letter's shape directly. Therefore, 26 classes of human gestures represent 26 English letters. All the modules and the experimental system given in Fig. 1 are described in the following sub-sections:

### *2.2. System Configuration*

Fig. 2 illustrates the entire experimental system. The hardware system consists of an industrial robotic arm, a Kinect device, a PC controller, and a writing board. The human demonstrator stands within the detection range of the Kinect device. The robotic arm system is placed in a fixed position facing the writing board. A controlling computer is used to control the Kinect and the robotic arm. During the experiments, the Kinect's sampling rate is approximately 30 frames per second, and its image output resolution is $320 \times 240$.

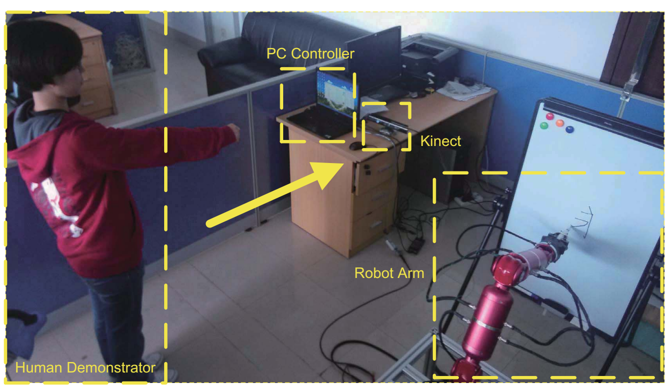


Figure 2: The experimental system consist of consists of an industrial robotic arm, a Kinect device, a PC controller, and a writing board.

Fig. 3 shows the layout of the demonstrator and the robot system. The robot system contains a five-DOF robotic arm. The arm is mounted in a fixed position on a frame, and four DOFs of the arm are applied to produce writing movements. A marker pen is mounted on the top of the robotic arm. The writing board is placed vertically within the arm's working range. This setup supplies the robot system with sufficient DOFs to act in 3-dimensional environments. In addition, each rotational motor of the robot arm contains a motor driver and an encoder that detects the motor's angle.

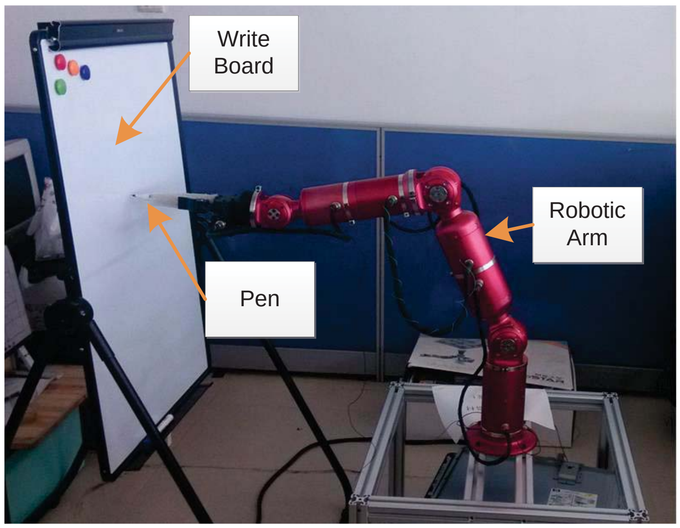


Figure 3: The robotic system and the writing board.

### *2.3. Gesture Sampling*

The gesture sampling module receives raw data from the Kinect device, filters the noise, and generates the captured trajectories for the remaining modules of the approach. The trajectory information is presented by the 3-dimensional trajectories of the human's right hand. Additionally, because two different behaviorial patterns are used, one to present Chinese characters and the other to present English letters, the font database is also implemented diversely.

The demonstrator performs a character and a letter by her right arm. Each Chinese character consists of a number of strokes, and each stroke of the character has its unique starting and end positions. In the control phase, the human demonstrator needs to provide the all the stroke information to the robot. Before performing a Chinese character's gestures, the human demonstrator needs to decide each stroke's size and position of the character, and then uses her right arm to perform all the strokes according to the character's writing sequence.

To write each stroke, the robot needs to move the pen over the stroke's starting position and then, to move the pen vertically until it touching the white board. After this, the robot horizontally moves the pen from the starting position to the end position, and subsequently uplifts the pen, moving it to the next stroke's starting position. The robot has to repeat the above procedure to complete all component strokes. To train the robot, the human demonstrator needs to provide it with the starting and end positions of each stroke and also, the stroke trajectory. The latter (stroke trajectory) is provided with the human hand's trajectory while keeping the arm straight. Therefore, only straight line gestures are correctly presumed.

In this work, the human demonstrator is requested to provide the relevant stroke information to the robot to facilitate learning. This includes the decision on each stroke's size and position regarding a given training character, and then the use of the demonstrator's right arm to show all the strokes according to the character's writing sequence. During this process, the human demonstrator needs to keep arm bent to move towards the next stroke's starting position; when the demonstrator considers that their hand has reached the correct position, the arm is straighten and ready to show the next stroke's trajectory.

### *2.3.1. Arm Configuration*

Therefore, only straight arm gestures (the pose in Fig. 4-B) are accepted and processed to generate the font trajectories. In order to start a new stroke or letter in a different position, while the arm is moving to a new starting position, the demonstrator must bend her arm (the pose in Fig. 4-A). Therefore, a detection algorithm is built to determine whether the arm is bent or straight.

Fig. 4 illustrates the arm gesture configuration. The picture is mirrored by the Kinect: the human's left arm in the figure is actually her right arm. The solid lines drawn on the demonstrator's arm represent the arm's skeleton information. "$L_1$" denotes the distance between the demonstrator's right shoulder and right

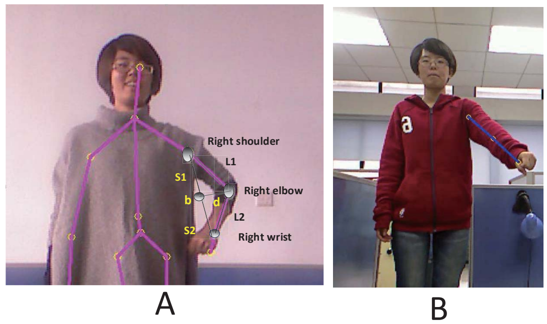


Figure 4: The gesture configuration. Panel A shows the bent arm gesture and Panel B shows the straight arm gesture.

elbow. "$L_2$" denotes the distance from the right elbow to the right wrist. The dashed line between the demonstrator's shoulder and right wrist is an auxiliary line. Point "b" is a floating point within the auxiliary line; its position is determined in an appropriate ratio based on the lengths of "$L_1$" and "$L_2$". "$S_1$" and "$S_2$" denote the distances from the human's right shoulder to point "b" and from point "b" to the wrist, respectively. Thus, the position of point "b" can be obtained by the following equations:

$$\gamma = \frac{L_1}{L_1 + L_2} \tag{1}$$

where, $\gamma$ is the distance proportion of "$S_1$" and "$S_2$", the proportion is determined by the distances of "$L_1$" and "$L_2$".

The position vector of "b" in the $x$, $y$, and $z$ axes are defined by:

$$\overrightarrow{b} = \overrightarrow{shoulder} + (\overrightarrow{wrist} - \overrightarrow{shoulder}) \cdot \gamma \tag{2}$$

where: $\overrightarrow{shoulder}$ and $\overrightarrow{wrist}$ denote the position vectors of the right shoulder and wrist, respectively. Thus, by using Eq. 2, the $x$, $y$, and $z$ coordinates of point "b" is determined. After that, the distance $d$ between point "b" and the right elbow is also determined by using Euclidean distance calculation.

In particular, the human demonstrator needs to keep the arm bent to move to each stroke's starting position. During the moving process, several bent arm gestures might be incorrectly recognized as the straight state. Such incorrect recognitions also exits in the straight arm's moving process. Therefore, another arm gesture, "slightly bent arm", is required to eliminate the incorrect recognitions. The angle of the slightly bent arm gesture is larger than that of the bent arm gesture, but less than that of the straight arm gesture. Once a gesture is recognized as the slight bent state, the sampling module will use the gesture's previous state as the arm's current state.

#### *2.3.2. Arm State Determination*

An arm state determination mechanism is required to check whether the demonstrator's arm is bent or straight. Our previous work on such mechanism [43] requires two thresholds, $\delta_{max}$ and $\delta_{min}$, to determine the arm's state. If $d$ is larger than $\delta_{max}$, the state of the right arm is bent. Else, if $d$ is less then another threshold, $\delta_{min}$, the state of the arm is straight. Otherwise, the state is regarded as a slightly bent state, then, the output of the mechanism is the arm's previous state. However, in this method, the values of $\delta_{max}$ and $\delta_{min}$ must be defined manually .

To avoid such human intervention and enhance the system's capacity, an Multilayer Perceptron (MLP) network is introduced in this paper to learn from sampled human gestures. Thus, the detection mechanism is learnt by the robot itself, rather than defined by human engineers. The network's input is the distance $d$ (the distance between point "b" and the right elbow); the network's output is the arm's status (bent, slight bent, or straight). 150 gestures, including 50 gestures of bent arm, 50 gestures of straight arm and 50 gestures of slightly bent arm are used to train the network.

The MLP network has one hidden layer with 20 hidden neurons and one output layer with three neurons. For this MLP network, implemented with the popular sigmoid activation function and the randomly generated initial weight, the conventional Back-propagation algorithm is used for training. The three output neurons indicates the bent, slightly bent and straight arm states.

### *2.4. Trajectory Optimization*

The Kinect device detects the human demonstrator's right hand position $hand(x, y, z)$. Because the robot writes characters and letters on the 2-dimensional writing board, the trajectory optimization module uses the $x$ and $y$ values only; thus, the depth value $z$ of the hand position is redundant in the work. However, during the phase while the demonstrator is gesturing characters or letters, the Kinect may fail to detect the hand's position. Therefore, a number of unexpected large range changes of the hand position may appear. The unexpected changes badly disturb the captured shapes of strokes and letters. These unexpected changes can be regarded as sharp pule signals. The amplitude-limiting filtering algorithm can simply filter such sharp pule signals.

Therefore, the amplitude-limiting filtering algorithm is used to filter out the unexpected changes. The distance $\omega$ between two consecutive positions ($p_{previous}$ and $p_{current}$) is calculated to determine whether an unexpected change takes place. If $\omega$ is less than a threshold $\delta$, the current position ($p_{current}$) will be maintained; otherwise, the next hand position is loaded as $p_{current}$. The entire method is demonstrated in the following pseudo code:

**Algorithm 1** The trajectory optimization procedure

Calculate the distance $\omega = \|p_{previous} - p_{current}\|$<br>
**if** $\omega < \delta$ **then**<br>
    let $p_{previous} = p_{current}$<br>
**else**<br>
    load the next point as $p_{current}$<br>
**end if**<br>
start a new iteration

## 3. Gesture Recognition and Robot Control

### *3.1. Gesture Recognition and Categories*

For writing Chinese characters, a set of five emblematic command gestures are chosen to present five elementary Chinese strokes. The five gesture examples are shown in the five panels of Fig. 5. The gestures are: (1) Horizontal stroke gesture (Panel A in Fig. 5); (2) Vertical stroke gesture (Panel B); (3) Left falling down stroke gesture (Panel C); (4) Right falling down stroke gesture (Panel D); and (5) Folder stroke (Panel E). Other research indicated that the fundamental Chinese strokes should have 28 types[44]. However, this paper can use the above five strokes to construct the 28 strokes. In addition, the paper will adopt a scale function to adjust each stroke's length; therefore, the basic five strokes are able to implement complex Chinese characters.

For gesturing these strokes, the demonstrator needs to perform as follows: (1) For the first stroke, raise the forearm to approximately shoulder height, then perform a horizontal waving motion. (2) For the second stroke, raise the arm to head height, and then wave vertically parallel to the body. (3) For the third stroke, raise the forearm towards head height, then push the hand downwards to the left side of the body. (4) For the fourth stroke, raise the forearm towards head height, then push the hand downwards to the right side of the body. (5) Regarding the fifth stroke, this gesture is a combination of the horizontal and vertical strokes.

For writing English letters, the human demonstrator merely straightens her arm, and moves her arm by following each English letter's shape. Two gestures are required to write the five letters, "f", "i", "j", "t", and "x". When the demonstrator finishes the first gesture, she must fold her arm and straighten it again to finish the rest of the gesture. In the experiment, time unit is used for gesture sampling. The demonstrator must gesture each character and letter within one time unit. The length of each time unit is two seconds. Note that: even a letter containing two gestures, the demonstrator also need finish gesturing in two seconds.

To train the classifier ensemble, the output of the training dataset module consists of a gesture's trajectory point vector $\overrightarrow{P}$ and the stroke or letter type $T_{s/l}$ that is assigned to the gesture. Hence, the data structure of the output is presented as $D_{training}(\overrightarrow{P}, T_{s/l})$. Each point vector $\overrightarrow{P}$ contains fifteen points, and each point is 2-dimensional with $(x_r, y_r)$ values. Note that $(x_r, y_r)$ values

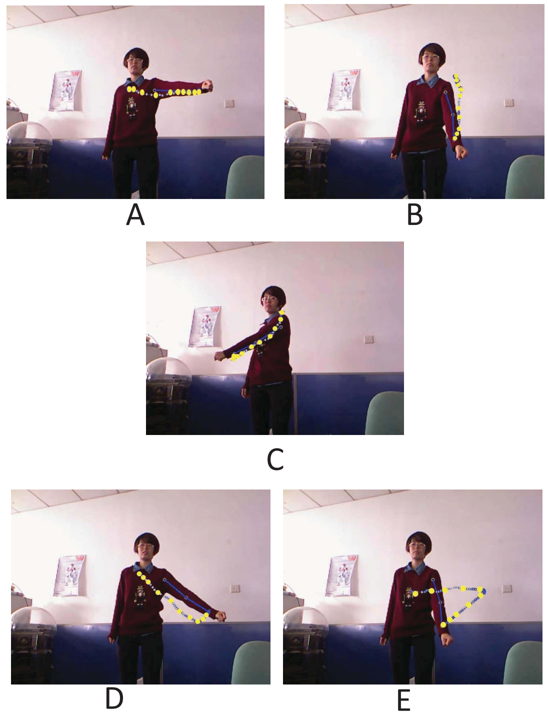


Figure 5: The examples of human gesture categories. Panel A: Horizontal stroke gesture; Panel B: Vertical stroke gesture; Panel C: Left falling down stroke gesture; Panel D Right falling down stroke gesture; and Panel E: Folder stroke.

indicate the relative position from the demonstrator's right hand to the right shoulder. $(x_r, y_r)$ values can be simply obtained by:

$$\begin{cases} x_r = wrist_x - shoulder_x \\ y_r = wrist_y - shoulder_y \end{cases} \tag{3}$$

Thus, the point vector has thirty dimensions. In addition, the stroke type contains only one element. Therefore, each training data $D_{training}(\overrightarrow{P}, T_{s/l})$ consists of 31 elements in total. Additionally, when the classifier has been built, the module's output will no longer contain the stroke label. Thus, the output during this phase is presented as $D_{working}(\overrightarrow{P})$, and has thirty elements. Both the Chinese and the English training data sets share the same data structure.

### 3.2. Font Database

The data structure of the font database is slightly different from the training pattern's. For writing English letters, the database requires the starting point position $x_s, y_s$ of each letter. Each point of the remaining fourteen points relies on its previous point. In other words, except for the first position, the remaining positions are relative values $(\Delta x_n, \Delta y_n)$ between the current position and the previous one. For example, the second position is the accumulation of the first position and the second position's relative value. Therefore, The English letter's data structure is presented as:

$$T_l : [(x_s, y_s), (\Delta x_1, \Delta y_1), (\Delta x_2, \Delta y_2), \cdots, (\Delta x_{14}, \Delta y_{14})]$$

where, $T_l$ is the letter type that is used as search index.

For writing Chinese characters, the data structure also adopts the same relative value structure as for English writing. However, in contrast to the English alphabet, one fixed type of Chinese stroke can differ markedly from one character to another. High quality Chinese writing cannot be simply achieved with only a single shape for each stroke. In this case, the five elementary strokes are implemented to have a scale function to construct various shapes of one type of stroke. Therefore, the database's input must include the starting position $x_s, y_s$ and the end position $x_e, y_e$ for each stroke. The stroke's data structure is presented as

$$\begin{aligned} T_s : [&(x_s, y_s), (x_e, y_e), (\alpha\Delta x_1, \beta\Delta y_1), (\alpha\Delta x_2, \beta\Delta y_2), \\ &\cdots, (\alpha\Delta x_{14}, \beta\Delta y_{14})] \end{aligned}$$

where, $T_s$ is the predefined stroke type that is used as the search index, $(x_e, y_e)$ denotes the end position of each stroke. $\alpha$ and $\beta$ are each point's scaling parameters defined by:

$$\begin{cases} \alpha = \dfrac{\|x'_s - x'_e\|}{\|x_s - x_e\|} \\ \beta = \dfrac{\|y'_s - y'_e\|}{\|y_s - y_e\|} \end{cases} \tag{4}$$

where: $x'_s, y'_s$ and $x'_e, y'_e$ are the new input stroke's starting and end positions, respectively, during the control phase. $x_e, y_e$ denote the end positions during the training phase. In this case, the human demonstrator uses different stroke's starting and end positions to control the stroke's shape.

### *3.3. Classifier Ensemble*

This module receives human gestures and produces the predictions of the gesture types. In terms of the system design, the gesture data contains 31 dimensions and 31 categories (26 letters and 5 strokes). A single conventional classifier may be unable to produce very accurate results. Therefore, In order to improve the prediction and reduce the robotic computational cost, in this work, a classifier ensemble method, developed in our previous work [34, 40], is applied to achieve better performance.

Before the ensemble produces predictions, a reduction process is invoked. This is because a classifier ensemble with a large group of classifiers can produce higher accuracy. However, eliminating redundant members can reduce the ensemble's complexity so as to save computational cost. Therefore, the fundamental concept and goals of ensemble reduction and feature selection are the same [34]. Each ensemble member is transformed into an "artificial feature", and such feature values are generated by collecting the respective classifier predictions. Feature selection algorithms can then be used to remove redundant features, so as to select a minimal classifier subset while preserving ensemble prediction accuracy.

In addition, because the harmony search algorithm exhibits a simplistic structure and powerful performance [45], it is applied to solve feature selection problems.

Fig. 6 illustrates the following four key steps of the classifier ensemble reduction approach used in this paper. Producing a diverse base classifier pool is the first step in producing classifier ensembles. Once the base classifiers have been built, the classifiers' decisions on the training instances are also collected. A feature selection algorithm is then performed on the collected data set to generate an optimal classifier ensemble. Then, the classifier is ready to recognize human gestures in the control phase. The details of the four key steps are

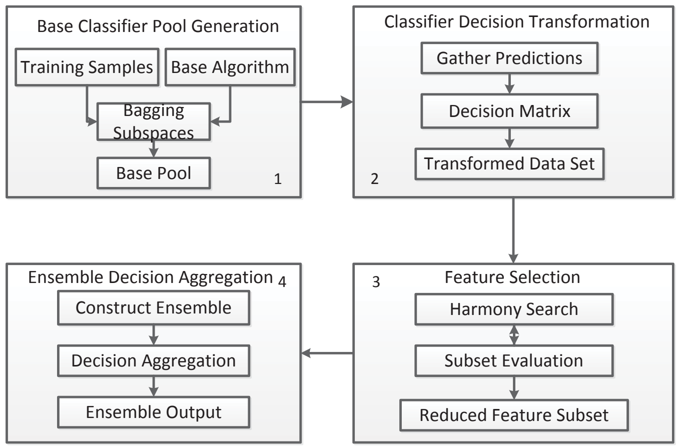


Figure 6: The flowchart of the classifier ensemble reduction. The flowchart consists of four steps.

described as follows:

**Base Classifier Pool Generation**: The first step is to form a diverse base classifier pool. The base classifier algorithm is C4.5. The conventional method – bagging algorithm – is used to build the base classifier pool. The ensemble diversity is achieved through selecting classifiers from different schools of classification algorithms.

**Classifier Decision Transformation**: This step combines the trajectory training pattern $D_{training}(\overrightarrow{P}, T_{s/l})$ and the classifier decisions with the classifier's format. Once the base classifiers are built, the classifier decisions on the captured trajectories are also gathered. For supervised feature selections methods, a class label ($T_{s/l}$) is required for each type of trajectory, and each single classifier's decision of a training instance is retained with the class label. A new dataset is therefore constructed where each column represents an artificially generated feature, and each row corresponds to a training instance.

**Feature Selection**: A new feature selection algorithm "Feature Selection with Harmony Search"(HSFS) [36] is then performed on the artificial dataset, evaluating the emerging feature subset using the predefined subset evaluator. HSFS optimizes the quality of discovered subsets, while trying to reduce subset

Table 1: Harmony Search Parameters

| $HMS$ | $Mu$ | $HMCR$ | $K$ |
|---|---|---|---|
| $10-20$ | 25 | $0.5-1$ | 1000 |

sizes. When the harmony search algorithm terminates, the best harmony is translated into a feature subset and returned as the feature selection result.

**Ensemble Decision Aggregation**: When the classifier ensemble is constructed, new gesture trajectories are classified by the ensemble members, and their results are aggregated to form the final ensemble decision output. The final aggregated decision is the winning classifier that has the highest averaged prediction across all classifiers.

In the training phase, this module will not provide any output, but only receives $D_{training}(\overrightarrow{P}, T_{s/l})$ from the Gesture Sampling Module. However, in the control phase, the module receives $D_{working}(\overrightarrow{P})$ and gives its prediction result $T(s/l)$ to the trajectory pattern module.

Table 1 gives the operating parameters of HSFS. In this study, HSFS applies four parameters: (1) the harmony memory size $HMS$, (2) the maximum number of iterations $K$, (3) the number of feature selectors $Mu$, and (4) the harmony memory considering rate $HMCR$. A parameter adjustment scheme for $HMS$, and $HMCR$ is used to dynamically change the two parameters. Please refer to [36] for further details of the dynamic change schema. In this paper, the value of $K$ is fixed as 1000 and the value of $Mu$ is set to 25.

### *3.4. Trajectory Pattern*

The trajectory pattern module uses, as indices to search the font database, the identified gesture type $T_{sl}$ that the classifier ensemble produces. The font database that contains the trajectory information of the English letters and the Chinese characters is established in the system's training phase. After searching, the trajectory pattern module produces the font's trajectories that correspond to the gesture types. The trajectories are then delivered to the robotic control module to drive the robot.

Because the gesture trajectory data supplied by the Kinect device is based on the Kinect's coordinate system, human gesture trajectories cannot be used as the robot control module's input directly. The following equation is applied to complete the conversion:

$$\begin{cases} p_x = 700 \\ p_y = 200 \cdot (x_s + \sum_{n=1}^{14} \Delta x_n) \\ p_z = 200 \cdot (y_s + \sum_{n=1}^{14} \Delta y_n + \psi) \end{cases} \tag{5}$$

where: $x_s$ and $y_s$ are the stroke's starting position; $\Delta x_n$ and $\Delta x_n$ are the subsequent relative points of the stroke; $n$ denotes the $n$th point; $p_x$ is the distance between the drawing board and the robot base; $p_y$ is the horizontal position of the drawing pen; $p_z$ indicates the the vertical position of the drawing pen. $\psi$ is an offset parameter to adjust the robotic writing position; the value of $\psi$ is set to 1.

By using Eq. 5, each human gesture is converted from a two-dimensional array to a three-dimensional array of points that is sent to the robotic arm control module. However, the robot motors executes joint values only. Therefore, the inverse kinematic algorithm is applied to transform the array to the robotic arm's joint values. The transformation and the control algorithm are introduced in the following section.

When the robot starts to write a new stroke or the second trajectory of the five letters consisting of two trajectories, the pen must be disengaged from the writing board. The disengagement movements are easily generated by setting a "standby" position. The position's $p_y$ and $p_z$ values are not changed; only the $p_x$ is given a $-200$ value. The procedure for writing one entire character is as follows: The robot writes the character's first stroke and, when finished, moves to the standby position. The robot then writes the character's second stroke; and again when finished, moves to the standby position. The robot continues this cycle until all the character's strokes are written.

More details on the robotic control systems are specified in the Appendix section.

## 4. Experimental Results and Analysis

Because the approach, as mentioned in Fig. 1, consists of the training phase and the control phase; the experimental procedure is also divided into the following two parts: (1) Classifier ensemble training part, and (2) Human gesture guided robotic writing part. In Part 1, four persons join the experiment to perform 31 predefined gestures. Each person performs about 70 times for each type of gestures. The captured dataset is used to train the classifier ensemble. In Part 2, only one human demonstrator stands in front of the Kinect device to perform Chinese characters and English letters with her right arm. The demonstrator is not one of the four persons in Part 1; therefore, 5 persons are involved in the experiments. Once it receives the stroke's trajectories, the robotic arm starts to move. Because the robot's writing speed is slower than that of the human's gesturing, the robotic writing actions have a very short delay. Usually, when the demonstrator finishes performing a character or letter's strokes, the robot still requires a bit of more time to complete the writing.

Based on the experimental procedure, the performance of the classifier ensemble is first tested after gathering the human's gesture. Then, the font database is evaluated to check whether the approach can retain correct English letter and Character stroke information. Next, the robotic arm system is enabled to write actual Chinese characters and English letters on the writing board. Two simple and one complex Chinese characters, the 26 English letters, and three English words are used to evaluate the entire approach. Each evaluation is illustrated in the following sections.

### *4.1. Classifier Ensemble Performance for Gesture Classification*

Table 2 lists the number of samples of the five elementary stroke gestures and the 26 English letter gestures. In order to extend the classifier's generalization,

Table 2: Number of Sampling Gestures

| Strokes | Horizontal | Vertical | Left Falling | Right Falling | Fold |
|---|---|---|---|---|---|
| Instances | 310 | 317 | 300 | 319 | 306 |
| Letters | “a” | “b” | “c” | ... | “z” |
| Instances | 300 | 300 | 300 | ... | 300 |

the gestures are performed by four different persons. Each category consists of more than 300 samples. Hence, the entire training dataset contains more then 9,300 samples in total [1]. In fact, the ensemble learning process might not require such a large number of training samples, using more than 300 samples for each class is to improve the classification accuracy.

To demonstrate the capability of the proposed CER framework, a number of experiments are conducted. The main ensemble construction method adopted is the bagging approach, and the base classification algorithm used is C4.5. The correlation-based feature selection algorithm (CFS) is employed as the feature subset evaluator. The HSFS algorithm then works together with the various evaluators to identify quality feature (classifier) subsets. In order to show the scalability of the framework, the base ensembles are created in three different sizes, 50, 100, and 200.

Table 3 summarizes the five sets of resulting classifiers (CFS, Random, Full, C4.5, and a MLP neural network). Several general observations can be drawn across all four setups. The prediction accuracies of the constructed classifier ensembles are universally superior to those achieved by a single C4.5 classifier and a MLP network. The accurate rates of the three types of C4.5 and MLP are less than 75%. For the ensemble of size 50, 100, and 200, CER with CFS achieves the highest accuracy among the five types of classifiers. Although the randomly formed ensemble are manually set to be of similar size to that used by CFS (e.g. when CFS is set to 41.7, 60.6, and 78.6, Random is set to 40, 60,

[1] The dataset's link: `https://www.dropbox.com/s/fb75rn16twm43p4/gesturedata.zip?dl=0`

and 80, correspondingly), the accuracy of Random is still worse then that of CFS. The full base classifier pool ensemble has similar performance with CFS; however, the ensemble size is much larger than that of CFS. A larger ensemble size tends to require a larger computational cost. Therefore, CER with CFS is the best choice for this robotic calligraphy system.

This result confirms the benefit of employing classifier ensembles. The result also demonstrates substantial ensemble size reduction, showing clear evidence of dimensionality reduction. Based on Table 3, in order to use the smallest ensemble size to achieve relatively good performance, the ensemble size is set as 40.

Table 3: C4.5 based ensemble classification accuracy result comparison

| | CFS | | Random | | Full | | | |
|---|---|---|---|---|---|---|---|---|
| Pool Size | Acc.% | Size | Acc.% | Size | Acc. | Size | C4.5 | MLP |
| 50 | **87.35** | 41.7 | 86.56 | 40 | 87.29 | 50 | | |
| 100 | **87.58** | 60.6 | 87.08 | 60 | 87.50 | 100 | 74.48 | 74.76 |
| 200 | **87.76** | 78.6 | 87.13 | 80 | 87.49 | 200 | | |

The confusion matrix in Fig. 7 illustrates the distributions of the classifier ensemble's errors. In this paper, the 10-fold cross-validation method is used to train the ensemble candidates, while another testing dataset, performed by the fifth demonstrator, is used to generate the confusion matrix. In the dataset, each class contains 50 samples. Both Axis $x$ and $y$ are the 26 English letters and the five elementary strokes. Therefore, the confusion matrix contains $31 \times 31$ grids. Labels "a - z" stands for the 26 English letters, and Labels "H", "V", "L", "R", and "F" stands for the horizontal, vertical, left falling, right falling, and folder strokes.

The color depth of each grid represents how many instances are classified correctly. Fig. 7 shows that almost all the instances are classified correctly. However, a small number of "j" instances are recognized as "i", and several

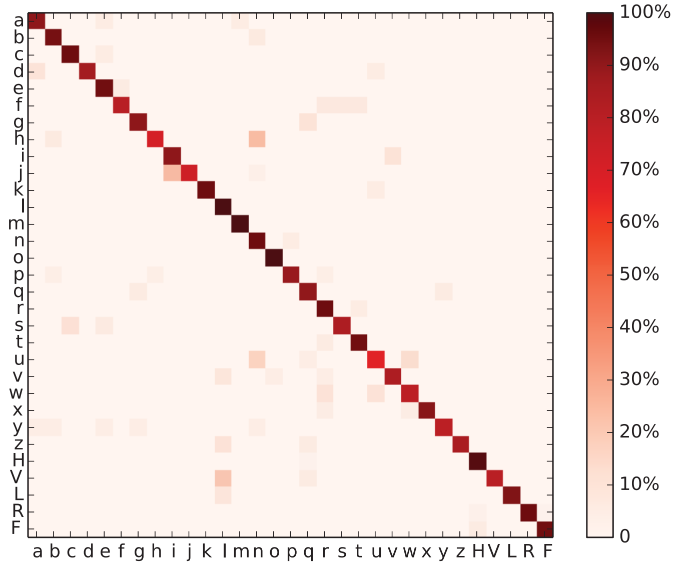


Figure 7: The confusion matrix of the strokes and letters.

"h" instance are incorrectly placed into the "n" category. Moreover, several "vertical strokes" are recognized as "l". Therefore, in order to avoid these types of mistakes, the demonstrator must perform the "j" and "h" letters slowly and carefully; in addition, the demonstrator must use markedly different gestures to assign the letter "l" and the vertical stroke.

In addition, the performance of deep neural networks is usually much better than that of the conventional classifiers. In the paper, C4.5 is chosen as the base classifier, the applied ensemble framework can select a number of base classifiers to form a classifier ensemble. Table 3 proves that the ensemble's performance usually is better than that of the single base classifier. Also, if a deep neural network is used as the base classifier, after the ensemble process, an ensemble of deep neural networks is obtained. We believe that such ensemble can also have a better performance than that of a single deep neural network.

*4.2. Gesture Sampling Results of the Font Pattern*

Fig. 8 presents the sampling result of the 26 English letters. The letters are chosen randomly from the training dataset. The trajectories of the letters are optimized by the noise filter algorithm: as such, the shapes are smooth enough for the robot to write. In particular, the output shapes of the letters "a, e, f, q, s, t, y" are clearly recognisable and resemble their block letter shapes. However, the "j, l" letters are not recognized easily. Letters of poor quality may be replaced, at human discretion, by choosing another instance of the same type from the training dataset. The poor quality letter can also be discarded from the training dataset at this point.

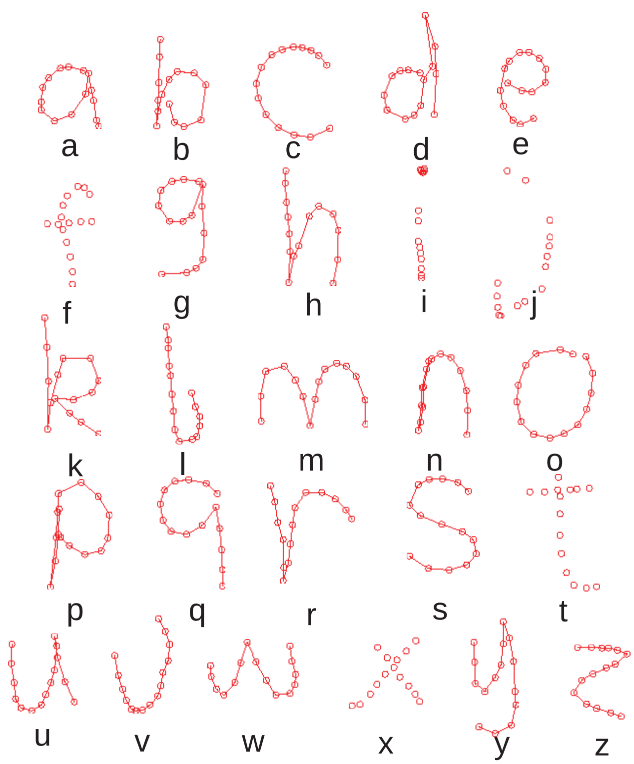


Figure 8: The sampling results of the 26 English letters.

To show the sampling differences among human demonstrators, more sampling results are presented in Fig. 9. The figure shows the sampling results of five letters: "a", "l", "f", "t", and "z". Each letter has five samples, which are generated by different human demonstrators. For each letter, the sampling results are slightly different from each other (although the differences are not large). If a human user is not satisfied with a sampling result, the user can select another sample result for the robot to write.

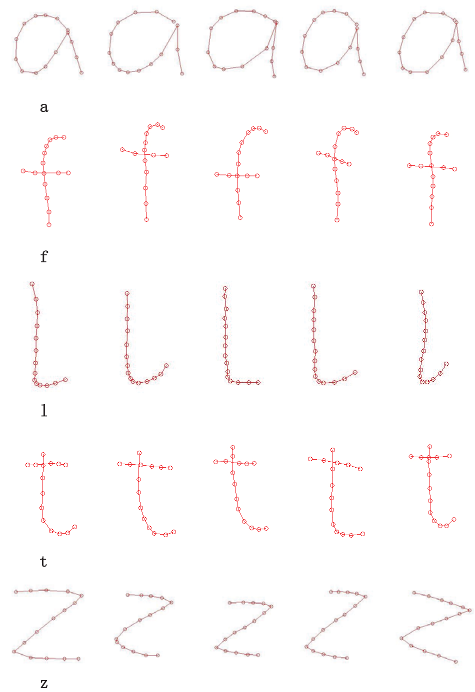


Figure 9: Different sampling results of five English letters.

Fig. 10 demonstrates the sampling result of the five elementary Chinese strokes. Standardised print versions of these strokes are presented alongside the captured samples, to the left. Note that although the resulting samples are correctly recognisable, they are unlikely to match the print templates precisely. The output depends instead on the demonstrators' versions during the training module. Herein lies the novelty of this approach, wherein the robotic system can capture and reproduce human expertise: for example, future medical surgical applications are likely to based off of human surgical motions, which cannot have any “standard” template.

The flexibility between untrained users are not evaluated in this paper. However, based on our previous research on classifier ensemble [34], one significant benefit is that the generalization ability is better than that of single classifier. In this case, an untrained user might also own high recognition accuracy. After testing the performance of the classifier ensemble and the font pattern, the experiment switches to Part 2.

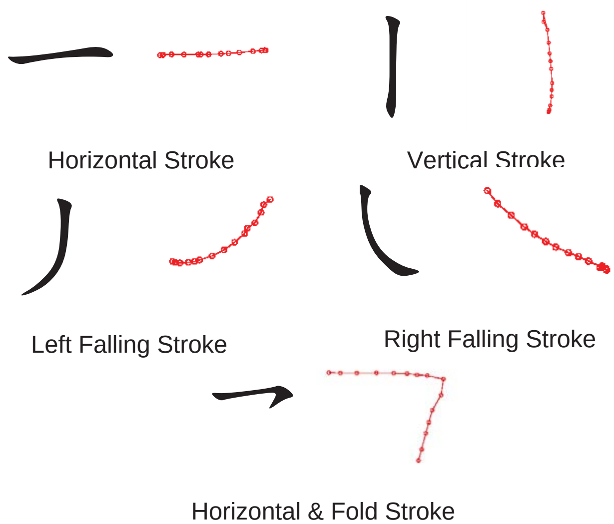


Figure 10: The writing results of the 5 elementary strokes.

## *4.3. Robotic Writing Results*

Fig. 11 illustrates the robotic writing results of the 26 English letters. All the letters are easily recognised. Comparing with Fig. 8, where the writing shapes basically follow the captured trajectories, only the letter “w” displays a mismatched appearance. In particular, the starting position is slightly different from the the captured trajectory. This situation may be caused by errors in the inverse kinematic calculations, or caused by the robotic arm’s motion accuracy.

Fig. 12 demonstrates the robot’s writing results of the five elementary strokes. To evaluate the writing results, a simple wavelet transform evaluation

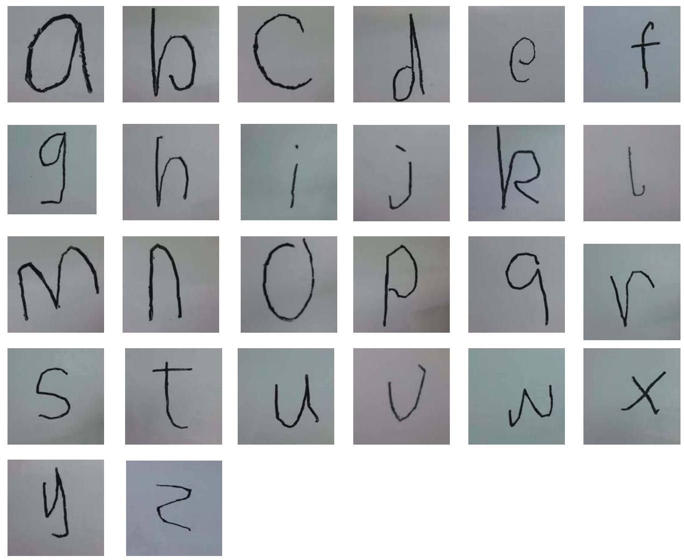

Figure 11: The robotic writing results of the English letters.

method [46] is applied. The evaluation results are used to present the writing quality. The writing quality is presented as scores, from 0 to 99, where a high score denotes better quality and low scores denote poor quality. The sampling results of the five strokes are listed on the left side of the writing results. The shape of the writing results are very similar to the sampling shapes. In addition, the scores of the horizontal, vertical, and fold strokes range from 85 to 90. Therefore, the writing quality of such strokes are satisfactory. However, the quality of the left falling and the right falling strokes is low: the scores of the two strokes are around 60. This low quality writing might be solved by adding more sampling points for the Chinese strokes.

Fig. 13 shows the writing results of two Chinese characters. The first character contains seven strokes, and the second character has five strokes in total. The characters in the left column of the figure are the printed style of the two Chinese characters. The robot generates the characters in the right column. Because these two characters are assembled by relatively simple strokes, the Kinect device can easily capture the human gestures.

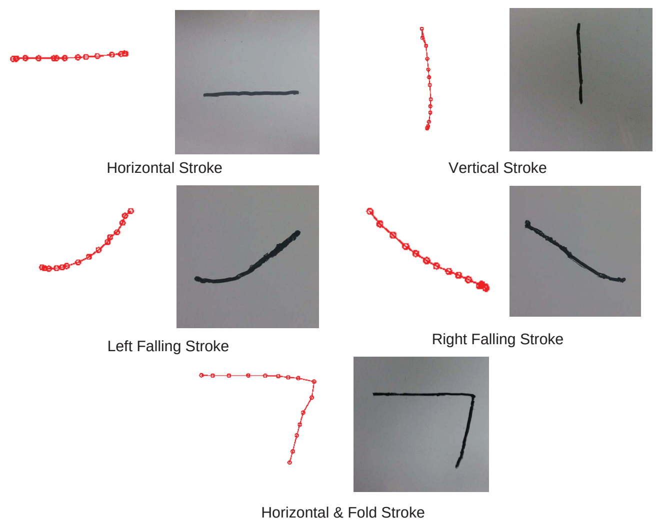


Figure 12: The sampling results of the 5 elementary strokes.

For those sequential gesture trajectories, the classifier ensemble algorithm also produces high recognition accuracy; except for the first stroke (labeled as ①) of Character 1, the remaining strokes of the two letters are recognized correctly. The correct type of Stroke ① is the left falling; however, it is recognized as a vertical stroke. This situation also occurred in Fig. 7, the proposed CER method has a few prediction errors, especially for the Chinese vertical stroke. The overall Chinese character is however recognisable: indeed even in day-to-day writing, one or two wrong strokes may not destroy an entire character. Moreover, because it is difficult to use the elementary five stroke to construct Stroke ② of the print version, the shapes of Stroke ② in the left and middle columns are not identical. This special stroke is replaced by the vertical stroke in the experiments.

In Character 2, the trajectory of Stroke ③ is not straight. The reason is that the scaling function of the font pattern can enlarge the size of strokes; meanwhile, small errors are also enlarged. The enlarged errors result in the non-straight trajectory of Stroke ③. In contrast, Stroke ④ is well written. In fact, Stroke ④ is an enlarged fold stroke. In addition, the layouts of the strokes of both characters are exactly correct. In the second character, especially, the starting and end positions of each stroke almost match. This indicates that the approach's trajectory conversion algorithm works properly.

Also in Fig. 13, Stroke ① in Character 1 is slightly different from Stroke ③ in Character 2. However, based on the proposed approach, the two strokes should be identical. The difference is caused by the robotic arm's repositioning accuracy. In addition, the writing board is not strictly vertical, this may also bring a little distortion to the strokes.

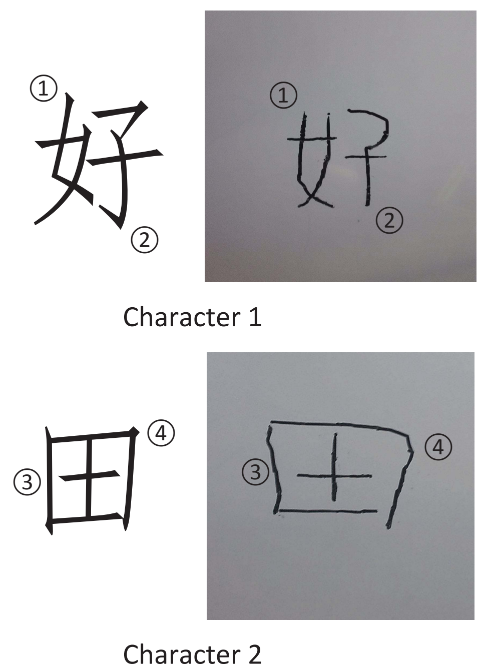


Figure 13: The writing results of two simple Chinese characters.

Fig. 14 shows the writing progress and result of a complex Chinese character "Yong". In the Chinese calligraphy, the "Yong" character is very significant to calligraphy beginners. Although the character consists of only eight types of strokes, these eight are the most important elementary strokes that can construct almost all Chinese characters. Therefore, if the robot can write the "Yong" character, in theory, the robot will be able to write all the Chinese characters. In the experiment, the “Yong” is simplified into six strokes. Step 1 is to write a right falling stroke; Step 2 is a fold stroke, Step 3 is a horizontal stroke, Step 4 is a left falling stroke, Step 5 is another left falling stroke, and Step 6 is a right falling stroke. Each stroke is correctly recognized, and its writing effect is good. The writing result of the entire “Yong” character (the last picture in Fig. 14) is excellent; the topological structure of the character is accurate; each stroke’s position is also accurate. Based on the figure, the robot’s arm can produce high quality writing and has the potential to write complex Chinese characters. This is evidence of how the proposed methods may be adapted to tackle complex multi-part robotic actions, for example the industrial welding of non-standard shapes.

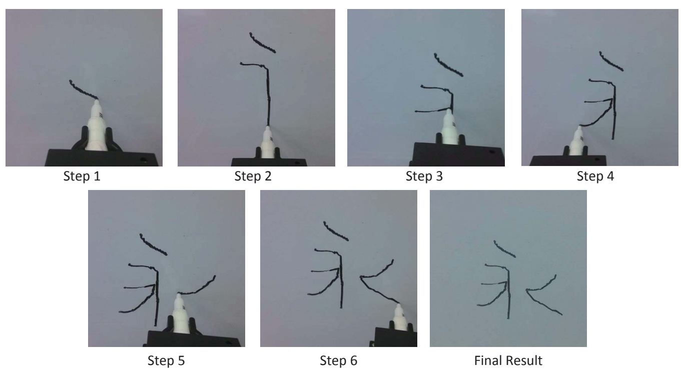


Figure 14: The robotic writing results of the typical Chinese characte.r

Fig. 15 shows the robotic writing results of three English words “let”, “big”, and “tea”. For each word, the demonstrator sequentially performed the three letters. For the output, the starting positions of the letters must be different; otherwise, the letters will overlap each other. The three words are written clearly and are recognizable. Because the robot invokes the saved human gestures as the font trajectory information, the shape of the letters are very close to the writing results presented in Fig 11. However, the size of the letters are changed a little bit, and is determined by how large the demonstrator writes. The results prove that the robotic writing system can not only write letters, but also write

English words. [46]

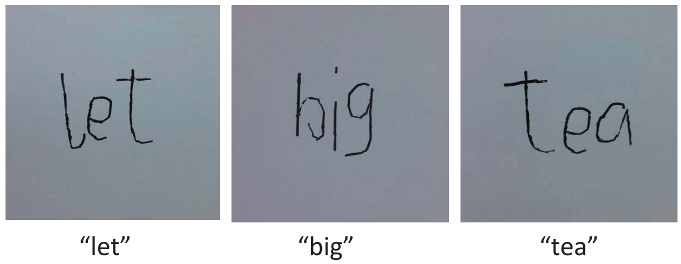


Figure 15: The robotic writing results of three English words.

In summary, the robotic system's classifier ensemble module succeeds in recognising the human demonstrator's gestures, the font database module correctly saves the trajectory information that has been performed by the demonstrator, and the font pattern generates accurate letter and character trajectories. The approach allows humans to conveniently control a robot to write many English words and Chinese characters.

### *4.4. Discussion and Comparison*

Based on the above experimental observations, the proposed approach is successful in generating robotic writing. This work significantly differs from existing work: our robot uses human gestures as the basis to collect trajectory information of English letters and Chinese characters. The classifier ensemble algorithm is also a novel and effective way for the robot to accurately recognise the human gestures. The scaling function of the font database leads the robot to write different Chinese characters by using the five elementary strokes.

To further reflect the strengths of this research, a comparison with typical robotic writing approaches is summarised. In particular, the comparison is focused on the following three important features: 1) font acquirement method, 2) system expansibility, and 3) multiple language support. The comparison is summarized as follows:

- **Font acquirement method**: Many existing approaches prefer to apply a direct programming method to implement font databases for robotic writing. However, this paper's approach is to use human gestures to obtain trajectory information. Similar to the pen-tablet method, this human gesture guided method also brings personal style into robotic writing; thus, the robotic writing can exhibit various styles, rather than exhibiting one fixed method. In future work, as an extension to the proposed system, the human gesture input can be conveniently replaced by EEG or EMG signals. Therefore, by involving the Brain-Computer Interface, our system can be extended to implement the "human consciousness controlled robotic writing" task, in other words: mind control.

- **System expansibility**: The overall robotic writing ability can be developed incrementally. Suppose the robotic system lacks letters or characters that are not already in the database, e.g. capital English letters. Conventional methods require human engineers to rebuild the font database through further programming. In contrast, our approach already contains a training and control phase; our approach merely switches to the training phase, and a human demonstrator performs new fonts in front of the Kinect device. After that, the robot has obtained the new trajectory information.

- **Multiple language support**: Another point to note is that existing methods do not show multiple language writing ability. Our approach enables the robot to write both English letters (linearly) and Chinese characters(inherently non-linear) by using one system. Because of the powerful performance of the classifier ensemble, developing further language writing ability is transformed into adding more instances to the classifier ensemble. Also, the human-interaction method reduces the work complexity of implementing the new language font.

To further reflect the strengths of this research, a comparison with the state of the art methods with gesture recognition is summarised in Table 4. The objectives of those methods are not the same, and therefore these state of the art methods do not form one unified dataset as a benchmark. However, in terms of the dataset size and gesture category, several findings can be identified. For example, the accuracy performance of the proposed approach (the first row in the table) is not the best; however, the recognisable gestures is the largest by far. More gesture categories lead to higher classification difficulty. Therefore, the classification performance of the proposed method is good enough.

## 5. Conclusion

This paper presented a human gesture guided approach to robotic handwriting. The approach first used the shapes of the English 26 letters and 5 Chinese strokes to define human gesture trajectories, and then applied a novel classifier ensemble to recognize human gestures. A 3D vision sensor, Kinect, is used to detect human arm motion trajectories, and the feature selection with a harmony search method is deployed to optimize the recognition accuracy. Moreover, a noise filtering program, a stroke scaling function and inverse kinematic calculations are implemented to obtain robotic motor values. The experimental results show that using human gestures can conveniently transform Chinese characters and English letter trajectory information from human right arm motion to the robot writing system; the classifier ensemble can recognize the human gestures with high accuracy; and the robotic system can easily write many Chinese characters and English words without complex programming and image processing work. In addition, the overall output writing quality is good; especially, each Chinese stroke's position is exactly correct and each letter of English words is clearly written.

While the proposed technique is promising, there is room for improvement. Firstly, the current system places a number of restrictions on the human, for example, the gesture trajectories are constrained in a 2D plane and the human wrist information is ignored. Therefore, we plan to add a pressure sensor or a visual feedback to adjust 3D positions, and use wrist information to control the robotic arm's movements. Also, the system defined a sample time limitation for human users, such limitation is not convenient; therefore, we plan to adopt left

| **Method** | **Data Source** | **Dataset Size** | **Gesture Category** | **Accuracy** |
|---|---|---|---|---|
| Reduced Classifier Ensemble | Captured by Kinect | 9352 | 31 | 87.43% |
| Human gesture corpora based method [27] | ChAirGest dataset | 1200 | 10 | 93.3% |
| Gaussian mixture modeling [29] | KTH dataset | 2391 | 6 | 98.3% |
| 3D extremity movement observation [30] | HumanEva-I Dataset | 800 | 6 | 93.1% |
| Active Learning of Ensemble Classifiers [33] | HumanEVA dataset | 800 | 9 | 86% |
| Hidden Markov Models [32] | HumanEVA dataset | 800 | 9 | 89.8% |
| Dynamic Bayesian Networks [28] | Seven videos from seven subjects | 490 | 10 | 99.59% |

Table 4: The comparison of gesture recognition

hand gestures, voice commands, or EMG and EEG signals to solve this problem. Furthermore, the median filter is a better choice to replace the amplitude limiting filter, so as to obtain smoother trajectories. Finally, the font information database contains only one type of handwriting style; if a human user performs dramatically different handwriting styles, our robot might not generate correct writing. With the font database increasingly being provided with more font information, the robot can gradually handle more different, and difficult, handwriting styles. The ability to automatically enrich the font database remains an important further work.

**Acknowledgement**

The authors are very grateful to the anonymous reviewers for their constructive comments that have helped significantly in revising this work. This work was supported by the National Natural Science Foundation of China (No. 61203336 and 61273338), the Fundamental Research Funds for the Central Universities the Fundamental Research Funds for the Central Universities (No. 20720160126), and the Major State Basic Research Development Program of China (973 Program) (No. 2013CB329502).

## Appendix

*Robotic Kinematics.* Fig. 16 demonstrates the robotic arm's configuration, which includes the setup of the arm's joints, links and orientations. The robotic arm has four links, labeled $l_1$, $l_2$, $l_3$ and $l_4$, with lengths of 150$mm$, 375$mm$, 354$mm$, and 175$mm$, respectively. Fig. 16 also indicates each joint's coordinate frame. The robotic arm's origin coordinate frame of the robotic arm is based on the first joint. The $x_0$ axis is vertical to the writing board; the $z_0$ axis is vertical to the ground, and the $y_0$ is vertical to the plane that is defined by the axes $x_0$ and $z_0$. Although the arm contains five rotation joints, four of which labeled $J_1$, $J_2$, $J_3$, and $J_4$ are applied for the writing system. In fact, three joints are sufficient for the robot to act in 3 dimensional workspace; however,

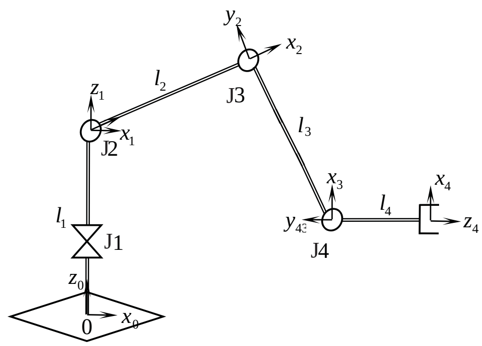


Figure 16: The configuration of the robot arm.

another joint is used to make the pen move vertically to the $z_0$ axis, thereby achieving high quality writing. The four joints work simultaneously to write each character or letter.

In order to control the robotic arm, the Denavit and Hartenberg (D-H) convention is used to analyze the direct and the inverse kinematics of the robot. Based on the coordinate systems illustrated in Fig. 16, the D-H parameter table is shown in Table 5. In the table, $\theta_i$ denotes the rotation angle from the $x_{i-1}$ axis to the $x_i$ axis along the $z_{i-1}$ axis; $d$ denotes the distance from the origin of the $(i-1)$ coordinate system to the intersection of the $z_{i-1}$ axis and the $x_i$ axis about the $z_{i-1}$ axis; $a$ denotes the distance from the intersection of the $z_{i-1}$ axis and the $x_i$ axis to the origin of the $i$th coordinate system about the $x_i$ axis; $\alpha$ denotes the rotation angle from the $z_{i-1}$ axis to the $z_i$ axis along the $x_i$ axis.

The rotation angle change occurs only in the first joint; thus, $\theta_1 = -90°$. In addition, the arm contains one more rotation joint, which is ignore in this work. Therefore, the ignore joint is regarded as a part of the link "$l_4$"; also, the joint is not included in the D-H parameter table. The working ranges of the four joints are $[-120°, 120°]$, $[-90°, 90°]$, $[-90°, 90°]$, and $[-45°, 45°]$.

The overall function of the motor system module is to produce the robotic arm's joint values; the input of this module is actually the pen's expected position. The values of $\theta_1$, $\theta_2$, $\theta_3$, and $\theta_4$ are obtained by using Eqs. 6, 7, 8 and 9.

Table 5: D-H parameter table

| **Joint No.** | $\theta$ | $d$ | $a$ | $\alpha$ | **Range** |
|---|---|---|---|---|---|
| 1 | $\theta_1$ | 0 | 150 | $-90$ | $[-120°, 120°]$ |
| 2 | $\theta_2$ | 0 | 375 | 0 | $[-90°, 90°]$ |
| 3 | $\theta_3$ | 0 | 354 | 0 | $[-90°, 90°]$ |
| 4 | $\theta_4$ | 0 | 175 | 0 | $[-45°, 45°]$ |

$$\theta_1 = \arctan \frac{p_x}{p_y} \tag{6}$$

$$\theta_2 = \arcsin - \frac{a_2 p_z + a_3 \cos\theta_3 p_z + a_3 \sin\theta_3 \sqrt{a_2^2 + a_3^2 + 2a_2 a_3 \cos\theta_3 - p_z^2}}{a_2^2 + a_3^2 + 2a_2 a_3 \cos\theta_3} \tag{7}$$

$$\theta_3 = \arccos \frac{(p_x \cos\theta_1 + p_y \sin\theta_1 - a_4)^2 + p_z^2 - a_2^2 - a_3^3}{2a_2 a_3} \tag{8}$$

$$\theta_4 = -\theta_2 - \theta_3 \tag{9}$$

The robotic control system is divided into two parts: (1) a hardware controller, and (2) a controlling computer. The hardware controller receives commands from the controlling computer, and converts the commands into the values that can be accepted by the robotic arm's motors. A driver program is built for communications between the computer and the hardware controller. The algorithm computer handles high level programs such as Kinect gesture recognition, a noise filter algorithm, classifier ensemble algorithm, and inverse kinematics calculations. The controlling computer uses a USB cable to connect the Kinect device and uses a CAN-Bus socket to communicate with the hardware computer. As for the software, the programs in the controlling computer are developed by using the C# programming language and the "Microsoft Kinect SDK 1.5"; the robotic control program is written in C++.

## References

[1] V. Potkonjak, Robot handwriting: why and how, in: A. Kecskeméthy, V. Potkonjak, A. Müller (Eds.), Interdisciplinary Applications of Kinematics, Springer, Netherlands, 2011, pp. 19–35.

[2] H. Zeng, Y. Huang, F. Chao, C. Zhou, Survey of robotic calligraphy research, CAAI Transactions on Intelligent Systems 11 (1) (2016) 15–26.

[3] Y. Man, C. Bian, H. Zhao, C. Xu, S. Ren, A kind of calligraphy robot, in: Proceedings of IEEE International Conference on Information Sciences and Interaction Sciences, IEEE Press, China, 2010, pp. 635–638.

[4] X. Ma, Q. Kong, W. Ma, X. Zhang, 4-DOF lettering robot's trajectory planning, Mechanical Engineering and Automation 165 (5) (2010) 161–163.

[5] B. D. Argall, S. Chernova, M. Veloso, B. Browning, A survey of robot learning from demonstration, Robotics and Autonomous Systems 57 (5) (2009) 469 – 483.

[6] F. Chersi, Learning through imitation: a biological approach to robotics, IEEE Transactions on Autonomous Mental Development 4 (3) (2012) 204–214.

[7] T. Liu, Y. Inoue, K. Shibata, Imitation control for biped robot using wearable motion sensor, Journal of Mechanisms and Robotics 2 (2) (2010) 024501–1–6.

[8] J. P. Bandera, J. A. Rodríguez, L. Molina-Tanco, A. Bandera, A survey of vision-based architectures for robot learning by imitation, International Journal of Humanoid Robotics 9 (1) (2012) 1250006–1–1250006–40.

[9] Z. Lu, X. Chen, Q. Li, X. Zhang, P. Zhou, A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices, IEEE Transactions on Human-Machine Systems, 44 (2) (2014) 293–299. doi:10.1109/THMS.2014.2302794.

[10] M. Holte, C. Tran, M. Trivedi, T. Moeslund, Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments, IEEE Journal of Selected Topics in Signal Processing 6 (5) (2012) 538–552. doi:10.1109/JSTSP.2012.2196975.

[11] M. A. Goodrich, A. C. Schultz, Human-robot interaction: A survey, Foundations and Trends in Human-Computer Interaction 1 (3) (2007) 203–275.

[12] A. Matsui, S. Katsura, A method of motion reproduction for calligraphy education, in: Proceedings of IEEE International Conference on Mechatronics, IEEE Press, Vicenza, 2013, pp. 452–457.

[13] C. Acosta-Calderon, H. Hu, Robot imitation: Body schema and body percept, Applied Bionics and Biomechanics 2 (3) (2005) 131–148.

[14] F. Chao, F. Chen, Y. Shen, W. He, Y. Sun, Z. Wang, C. Zhou, M. Jiang, Robotic free writing of Chinese characters via human robot interactions, International Journal of Humanoid Robotics 11 (1) (2014) 1450007–1–26.

[15] T. Kulvicius, K. Ning, M. Tamosiunaite, F. Worgötter, Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting, IEEE Transactions on Robotics 28 (1) (2012) 145–157. doi:10.1109/TRO.2011.2163863.

[16] A. Billard, S. Calinon, R. Dillmann, S. Schaal, Survey: Robot programming by demonstration, Handbook of Robotics, . chapter 59, 2008 (2008).

[17] R. Dillmann, Teaching and learning of robot tasks via observation of human performance, Robotics and Autonomous Systems 47 (2C3) (2004) 109 – 116, robot Learning from Demonstration.

[18] R. Dillmann, T. Asfour, M. Do, R. Jäkel, A. Kasper, P. Azad, A. Ude, S. Schmidt-Rohr, M. Lösch, Advances in robot programming by demonstration, KI - Knstliche Intelligenz 24 (4) (2010) 295–303. doi:10.1007/s13218-010-0060-0.
URL http://dx.doi.org/10.1007/s13218-010-0060-0

[19] F. Yao, G. Shao, J. Yi, Trajectory generation of the writing-brush for a robot arm to inherit block-style chinese character calligraphy techniques, Advanced Robotics 18 (3) (2004) 331–356.

[20] F. Chao, Y. Sun, Z. Wang, G. Yao, Z. Zhu, C. Zhou, Q. Meng, M. Jiang, A reduced classifier ensemble approach to human gesture classification for robotic Chinese handwriting, in: Proceedings of IEEE Conference on Fuzzy Systems (FUZZ-IEEE), 2014, pp. 1720–1727.

[21] G. Tsoumakas, I. Partalas, I. Vlahavas, A taxonomy and short review of ensemble selection, in: ECAI 2008, workshop on supervised and unsupervised ensemble methods and their applications, 2008, pp. 41–46.

[22] H. Zhou, H. Hu, H. Liu, J. Tang, Classification of upper limb motion trajectories using shape features, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (6) (2012) 970–982.

[23] V. Ramakrishna, T. Kanade, Y. A. Sheikh, Tracking human pose by tracking symmetric parts, 2013.

[24] M. Ramanathan, W.-Y. Yau, E. K. Teoh, Human action recognition with video data: Research and evaluation challenges, IEEE Transactions on Human-Machine Systems 44 (5) (2014) 650–663. doi:10.1109/THMS.2014.2325871.

[25] E. Ohn-Bar, M. Trivedi, Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations, IEEE Transactions on Intelligent Transportation Systems 15 (6) (2014) 2368–2377. doi:10.1109/TITS.2014.2337331.

[26] D. Gong, G. Medioni, X. Zhao, Structured time series analysis for human action segmentation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (7) (2014) 1414–1427. doi:10.1109/TPAMI.2013.244.

[27] S. Ruffieux, D. Lalanne, E. Mugellini, O. A. Khaled, Gesture recognition corpora and tools: A scripted ground truthing method, Computer Vision and Image Understanding 131 (2015) 72–87, special section: Large Scale Data-Driven Evaluation in Computer Vision. doi:http://dx.doi.org/10.1016/j.cviu.2014.07.004. URL http://www.sciencedirect.com/science/article/pii/S1077314214001568

[28] H. I. Suk, B. K. Sin, S. W. Lee, Hand gesture recognition based on dynamic bayesian network framework, Pattern Recognition 43 (9) (2010) 3059–3072.

[29] M. Vrigkas, V. Karavasilis, C. Nikou, I. A. Kakadiaris, Matching mixtures of curves for human action recognition, Computer Vision and Image Understanding 119 (0) (2014) 27–40. doi:http://dx.doi.org/10.1016/j.cviu.2013.11.007. URL http://www.sciencedirect.com/science/article/pii/S1077314213002270

[30] C. Tran, M. M. Trivedi, 3-D posture and gesture recognition for interactivity in smart spaces, IEEE Transactions on Industrial Informatics 8 (1) (2012) 178–187. doi:10.1109/TII.2011.2172450.

[31] S. Mitra, T. Acharya, Gesture recognition: A survey, IEEE Transactions on System, Man, and Cybernetics, Part C 37 (3) (2007) 311–324.

[32] J. Richarz, G. A. Fink, Visual recognition of 3D emblematic gestures in an HMM framework, Journal of Ambient Intelligence and Smart Environments 3 (2011) 193–211.

[33] J. Schumacher, D. Sakič, A. Grumpe, G. A. Fink, C. Wöhler, Active learning of ensemble classifiers for gesture recognition, in: A. Pinz, T. Pock, H. Bischof, F. Leberl (Eds.), Pattern Recognition, Vol. 7476 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 498–507.

[34] R. Diao, F. Chao, T. Peng, N. Snooke, Q. Shen, Feature selection inspired classifier ensemble reduction, IEEE Transactions on Cybernetics 44 (8) (2014) 1259–1268.

[35] G. Yao, F. Chao, H. Zeng, M. Shi, M. Jiang, C. Zhou, Integrate classifier diversity evaluation to feature selection based classifier ensemble reduction, in: 2014 14th UK Workshop on Computational Intelligence (UKCI), 2014, pp. 1–7. doi:10.1109/UKCI.2014.6930156.

[36] R. Diao, Q. Shen, Feature selection with harmony search, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42 (6) (2012) 1509–1523.

[37] G. Giacinto, F. Roli, An approach to the automatic design of multiple classifier systems, Pattern Recognition Letter 22 (1) (2001) 25–33.

[38] I. Partalas, G. Tsoumakas, I. Vlahavas, Pruning an ensemble of classifiers via reinforcement learning, Neurocomputing 72 (7–9) (2009) 1900–1909.

[39] F. Markatopoulou, G. Tsoumakas, L. Vlahavas, Instance-based ensemble pruning via multi-label classification, in: Proceedings of 22nd IEEE International Conference on Tools with Artificial Intelligence, Vol. 1, IEEE, Arras, 2010, pp. 401–408.

[40] G. Yao, H. Zeng, F. Chao, C. Su, C.-M. Lin, C. Zhou, Integration of classifier diversity measures for feature selection-based classifier ensemble reduction, Soft Computing 20 (8) (2016) 2995–3005. doi:10.1007/s00500-015-1927-7.
URL http://dx.doi.org/10.1007/s00500-015-1927-7

[41] A. Zaraki, D. Mazzei, M. Giuliani, D. De Rossi, Designing and evaluating a social gaze-control system for a humanoid robot, IEEE Transactions on Human-Machine Systems 44 (2) (2014) 157–168. doi:10.1109/THMS.2014.2303083.

[42] D. Cazzato, M. Leo, C. Distante, An investigation on the feasibility of uncalibrated and unconstrained gaze tracking for human assistive applications by using head pose estimation, Sensors 14 (5) (2014) 8363–8379. doi:10.3390/s140508363.

[43] F. Chao, M. H. Lee, C. Zhou, M. Jiang, An infant development-inspired approach to robot hand-eye coordination, International Journal of Advanced Robotic Systems 11 (15) (2014) 1–14.

[44] F. Yao, G. Shao, Modeling of ancient-style chinese character and its application to ccc robot, in: Proceedings of IEEE International Conference on Networking, Sensing and Control, IEEE Press, FL, 2006, pp. 72–77.

[45] B. Naik, J. Nayak, H. S. Behera, A. Abraham, A self adaptive harmony search based functional link higher order ANN for non-linear data classification, Neurocomputing 179 (2016) 69–87. doi:10.1016/j.neucom.2015.11.051.
URL http://dx.doi.org/10.1016/j.neucom.2015.11.051

[46] Y. Dong, The achievement of robot calligraphy evaluation system based on projected signal and wavelet transform, Master's thesis, Xiamen University (2014).