A Reduced Classifier Ensemble Approach to Human Gesture Classification for Robotic Chinese Handwriting

Fei Chao, Yan Sun, Zhengshuai Wang, Gang Yao, Zuyuan Zhu, Changle Zhou, Qinggang Meng and Min Jiang

***Abstract*— The paper presents an approach to applying a classifier ensemble to identify human body gestures, so as to control a robot to write Chinese characters. Robotic handwriting ability requires complicated robotic control algorithms. In particular, the Chinese handwriting needs to consider the relative positions of a character's strokes. This approach derives the font information from human gestures by using a motion sensing input device. Five elementary strokes are used to form Chinese characters, and each elementary stroke is assigned to a type of human gestures. Then, a classifier ensemble is applied to identify each gesture so as to recognize the characters that gestured by the human demonstrator. The classier ensemble's size is reduced by feature selection techniques and harmony search algorithm, thereby achieving higher accuracy and smaller ensemble size. The inverse kinematics algorithm converts each stroke's trajectory to the robot's motor values that are executed by a robotic arm to draw the entire character. Experimental analysis shows that the proposed approach can allow a human to naturally and conveniently control the robot in order to write many Chinese characters.**

I. Introduction

The robotic writing ability is an interesting research field, which focuses on how to automatically control robotic actuators to write complex characters from single strokes [1]. It is very difficult for robots to draw Chinese characters containing human-like handwriting features. This is because handwriting, being a typical human motion, is a highly demanding task regarding kinematics and dynamics[2]. Moreover, robotic writing needs to consider a redundant number of degrees-of-freedom (DOFs) [3]. Additionally, Chinese character writing is much more complicated than English writing, this is because robots may require to contain more DOFs and to perform more postures for writing Chinese characters. Therefore, writing Chinese can be used as a "test bed" to evaluate the flexibility and control methods of various robots.

Robotic writing Chinese characters requires robots to obtain character font information. A number of current approaches usually applied direct programming methods to embed fonts database inside robot's control systems. However, if human users draw a non-predefined set of objects such as an artistic free drawing, the current approach cannot support this ability. Therefore, imitation of human gestures supplies free drawing scenario for robotic writing. On the other hand, the imitation of human gestures reduces the enhancement complexity of robot's control algorithms. Because the imitation is considered as an effective learning method to transfer skills and knowledge from human beings to robots [4][5]. Especially, it is very crucial for robots to acquire new skills without the requirements of repeated training or complex programming [6]. Therefore, we believe that applying human gestures to control robotic writing will reduce the task's complexity. Also, human users can use a convenient and natural way to control robots to write the characters that human users demand to write.

In order to guide a robot to write Chinese characters by using human's gestures, in this paper we focus on the aspect of mapping the motion parameters of the relevant body parts, e.g. the gesturing hand or the upper body including shoulder and arms, to a gesture or action category. A number of different classifier methods to classify various types of gestural expressions – ranging from arm gestures to full-body motion – have been reported in the literature [7][8]. Several time-series data analysis methods, ie. "Dynamic Bayesian Networks" [9] and "Hidden Markov Models" [10][11] are usually applied to solve the problem of recognising variable length trajectories of human gestures.

However, to improve the recognition accuracy rate and to simply the recognition system, the work inspired by [12] addresses the problem by classifying trajectory segments comprising a fixed number of sampling points of a human gesture. In addition, the "Kinect" motion sensing input device is able to capture human body's skeleton information. The information is then converted to an array of hand trajectory data. Also, for the recognition accuracy, we make use of classifier ensembles for recognition, since the classifier ensembles are known to usually improve recognition performance in a wide range of pattern recognition tasks [13]. One Chinese character can be dissembled to a number of elementary strokes. In this study, 5 classes of human gestures are assigned to 5 types of strokes. A human demonstrator performs diverse poses to represent the corresponded strokes, so as to implement the entire character. A 3 DOF robotic arm receives the captured pattern, and kinematic algorithms are used to convert the stroke trajectories to the arm's joint values; and then, completes the writing task.

The rest of this paper is organized as follows: Section II briefly introduces the background of classifier ensemble and robotic writing. Section III describes the main implementation issues and an experimental robot system. Section

F. Chao, Y. Sun, Z. Wang, G. Yao, Z, Zhu, C. Zhou and M. Jiang are with the Cognitive Science Department, Fujian Provincial Key Laboratory of Brain-like Intelligent Systems, School of Information Science and Engineering, Xiamen University, P. R. China (email: fchao@xmu.edu.cn). Q. Meng is with the Department of Computer Science, Loughborough University, UK.

This work was supported by the National Natural Science Foundation of China (No. 61203336, 61273338 and 71301133) and the Major State Basic Research Development Program of China (973 Program) (No. 2013CB329502).

IV presents the experimental results and discusses their implications. Section V concludes the paper and points out future research.

II. Classifier Ensemble and Robotic Writing

The main purpose of a classifier ensemble is to improve the performance of single classifier system. However, an ensemble with too many classifiers may occupy a large number of computational time. The objective of classifier ensemble reduction (CER) is to reduce the amount of redundancy in a preconstructed classifier ensemble, to form a much reduced subset of classifiers that can still deliver the same classification results [7]. It is an intermediate step between ensemble construction and decision aggregation. Efficiency is one of the obvious gains from CER. Having a reduced number of classifiers can eliminate a portion of run-time overheads, making the ensemble processing quicker; having fewer models also means relaxed memory and storage requirements. Removing redundant ensemble members may also lead to improved diversity within the group, and further increase the prediction accuracy of the ensemble. Existing approaches in the literature include techniques that employ clustering [14] to discover groups of models that share similar predictions, and subsequently prune each cluster separately. Others use "Reinforcement Learning" [15] and "Multi-label Learning" [16] to achieve redundancy removal. In this paper, a new frame work for CER that builds upon the ideas from existing feature selection techniques [13]. Each ensemble member is transformed into an artificial feature in a data set, and the feature values are generate by collecting the classifiers's predictions. Feature selection algorithm is then used to remove redundant features.

The core technique of the robotic writing ability is transforming one character's strokes to robotic motor values; then, robotic actuators use the motor values to act. Therefore, the problem of robotic writing is required to understand how to acquire a character's font information. Several works used commercial front libraries directly [17], [18]. Other works applied image processing technologies to extract character's font information from copybooks or human's handwritings [19][20]. The above two methods have the advantage of using predefined font databases to plan a robot actuator's trajectories, but are not sufficiently flexible to generate the arbitrary curves needed in handwriting or drawing. In a number of studies, robots can only write Chinese character's strokes rather than write an entire Chinese character; these studies require human engineers to disassemble each Chinese character into different strokes, and assign the stroke positions to the robots to finish the writing [1][21][22][23][24][25]. These approaches can create high quality robotic writing; however, they are inconvenient in making the robots write new characters. Additionally, only a few works have used data gloves and brain-machine interface devices to collect writing gesture information, and then convert this information to robotic joint values [26][27][28]. This category of approach inspired us to consider if human gestures could directly control robots to write any characters that a human wants to write. However, the cost of such devices is high; therefore, we plan to find a cost-effective alternative "Kinect" to capture human gestures.

III. The Approach

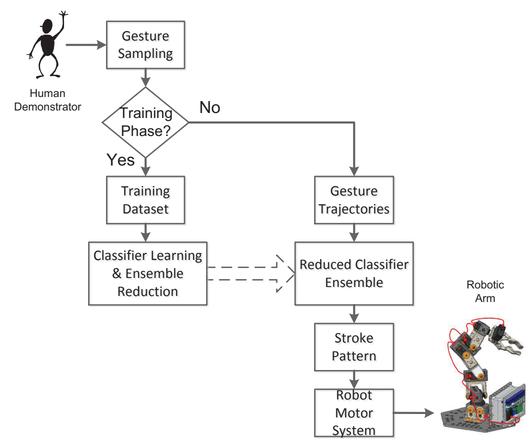

Fig. 1. The flowchart of the robotic handwriting

Figure 1 demonstrates the working flowchart of the robotic handwriting approach. The approach consists of two parts: 1) implement classifier learning and ensemble reduction; and 2) use the reduced classifier ensemble to identify the gestures that are performed by a human demonstrator and write the identified strokes. In the first part, a human demonstrator repeats to gesturing five predefined poses in front of the motion sensing input device "Kinect". Only the skeleton information of the human's poses are captured by the Kinect. Thus, the gestures are presented by 2 dimensional point arrays. These point arrays are kept into a temporary dataset; then, the dataset is applied to train a classifier ensemble. A new ensemble reduction method, implemented by feature selection and harmony search techniques, is applied to train the classifier ensemble, rather than conventional ensemble approach. After the training, a size reduced classifier ensemble with high recognition accuracy is obtained, and ready for the second part.

In the second part, the human demonstrator does not need to repeat to gesturing the predefined patterns. The demonstrator merely behaves each stroke's gesture in sequence. The Kinect device converts the gesture trajectories to skeleton data that are directly sent to the reduced classifier ensemble. Since the classifier ensemble has been trained in the first part, the ensemble is able to generate the stroke type that is correspondent to the gesture. The stroke pattern module transforms the stroke type and the stroke's position to the robotic arm's joint values. Then, the robot executes the joint values to write the stroke. All the modules listed in Figure 1 are illustrated in the following sub-sections.

A. Gesture Sampling

When the human demonstrator gestures a character by his/her right arm, only straight arm gestures are recognized and processed to generate the character's strokes. However, in order to start a new stroke in a different position, the demonstrator must bend his/her arm. Therefore, we build an algorithm to determine whether the arm is bent or straight. In addition, the demonstrator also controls the robot to dip in ink and drag the canvas, thus, several of the left arm's gestures are set to support the above two functions.

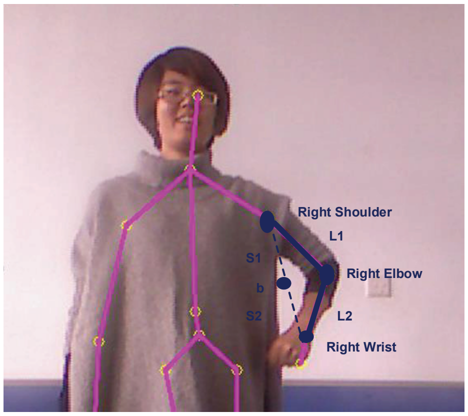

Fig. 2. The gesture configuration

Figure 2 illustrates the arm gesture determination algorithm. The picture is mirrored by the Kinect. The solid line with dark blue stands for the demonstrator's right arm. "L_1" is the distance between his/her right shoulder and his/her elbow. "L_2" is the distance from the elbow to the wrist. The dash line indicates the distance between the demonstrator's shoulder and the right wrist. "b" is a floating point within the dash line; its position is determined in an appropriate ratio based on the lengths of "L_1" and "L_2". Thus, we have:

$$b_i = shoulder_i + (wrist_i - shoulder_i)\frac{L_1}{L_1 + L_2} \tag{1}$$

where: i has 1, 2 and 3, which stand for the values in x, y and z axes; vectors $shoulder$ and $wrist$ are the positions of the shoulder and the wrist, respectively. Thus, by using Equation 1, we can obtain the x, y, and z coordinates of point b. Then, the distance d between point b and the right elbow is calculated by:

$$d = \sqrt{\sum_{i=1}^{3}(b_i - elbow_i)^2} \tag{2}$$

The value of d is used to determine whether the arm is bent or not. If d is less then a threshold δ_{min}, the state of the arm is straight. If d is larger than another threshold δ_{max}, the arm is bent. Otherwise, the state inherits the arm's previous state. Eq. 3 gives the demonstrator arm's state at the kth sampling:

$$state_k = \begin{cases} straight & \text{if } d < \delta_{min} \\ bent & \text{if } d > \delta_{max} \\ state_{k-1} & \text{otherwise} \end{cases} \tag{3}$$

where, in this paper, $\delta_{min} = 0.05$ and $\delta_{max} = 0.08$. The solution of the arm state's determination is not complicated, and it is suitable for any human demonstrator rather than a specific person.

The Kinect senses the right wrist position values $wrist(x, y, z)$ of the demonstrator. The trajectory capture module uses $wrist(x, y)$ values only, because our robot writes characters on the two-dimensional canvas; thus, the depth value is redundant. However, during the phase when the Kinect senses the wrist positions, the Kinect may lost the wrist; therefore, several unexpected large changes of the wrist positions may occur. These unexpected changes terribly disturb the stroke's shape. In this case, we apply the amplitude-limiting filtering algorithm to filter the unexpected changes. We calculate the distance $dist$ between two consecutive positions (pos_{pre} and pos_{cur}). If $dist$ is less than a amplitude $\delta_{amplitude}$, the current position will be kept; otherwise, the current position is discarded. The algorithm is illustrated in the following pseudo code:

Algorithm 1 The amplitude-limiting filtering algorithm

1: Calculate the distance $dist = \|pos_{pre} - pos_{cur}\|$
2: **if** $dist < \delta_{amplitude}$ **then**
3: let $pos_{pre} = pos_{cur}$
4: **else**
5: discard pos_{cur}
6: **end if**
7: go back to Step 1

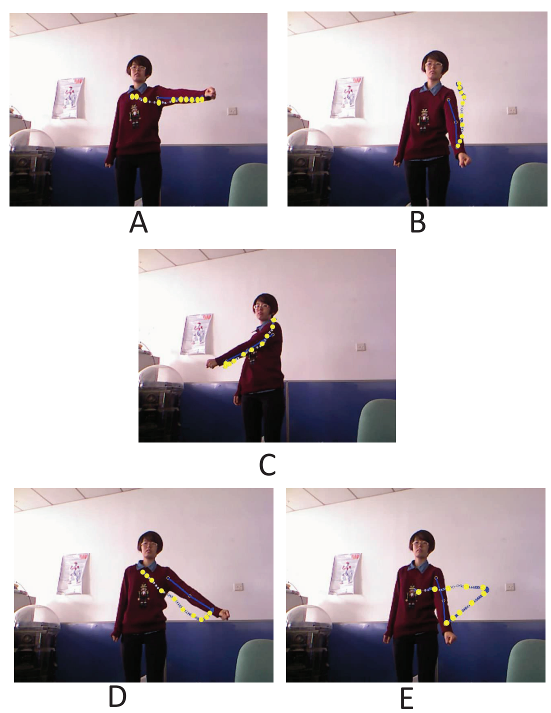

Fig. 3. The examples of human gesture categories

A set of five emblematic command gestures are chosen to present five elementary Chinese strokes. The five gesture examples are shown in Figure 3's five pictures. The gestures are: 1) Horizontal stroke gesture (Picture A in Figure 3): Raise the forearm to approximately shoulder height, then perform a horizontal waving motion. 2) Vertical stroke gesture (Picture B): Raise the arm to head height, and then wave vertically parallel to the body. 3) Left falling down stroke gesture (Picture C): Raise forearm towards head height, then push hand downwards to the left side of the body. 4) Right falling down stroke gesture (Picture D): Raise forearm towards head height, then push hand downwards to the right side of the body. 5) Fold folder stroke (Picture E): This gesture is a combination of the horizontal stroke and the vertical stroke. When the horizontal stroke has been gestured, vertically move the arm downwards.

For training the classifier ensemble, the output of this module consists of a gesture's trajectory point vector $\overrightarrow{P}$ and the stroke type T_{stroke} that is assigned to the gesture. Hence, the data structure of the output is presented as $D_{training}(\overrightarrow{P}, T_{stroke})$. Each point vector $\overrightarrow{P}$ contains 10 points, and each point is 2-dimensional with x, y. Thus, the point vector has 20 dimensions. In addition, the stroke type contains only 1 element. Therefore, each training data $D(\overrightarrow{P}, T_{stroke})$ consists of 21 elements in total. On the other hand, when the classifier has been built, the module's output will not contain the stroke label any more. Thus, the output during this term is presented as $D_{working}(\overrightarrow{P})$, and has 20 elements.

B. Classifier Ensemble Reduction using Feature Selection Techniques

The classifier ensemble receives human gestures, and gives the predictions of the gestures. Before the ensemble produces predictions, a reduction process is invoked. The feature selection technology is applied to handle the reduction process. This idea is based on the same concepts of classifier ensemble reduction and feature selection. In addition, harmony search algorithm exhibits simplistic structure and powerful performance, so that the algorithm is applied to solving feature selection problems.

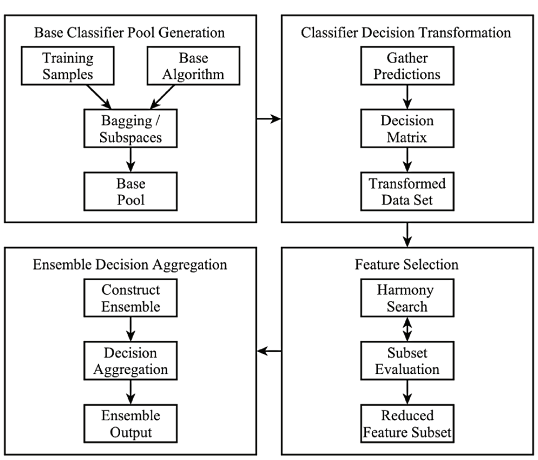

Fig. 4. The flowchart of the classifier ensemble reduction

Fig. 4 illustrates the four key steps of the classifier ensemble reduction approach used in this paper.

1) The first step is to form a diverse base classifier pool. Both bagging and random subspaces methods can be used to build the base classifiers. In this paper, a mixed classifier scheme is implemented. By selecting classifiers from different schools of classification algorithms, the diversity is naturally achieved through the various foundations of the algorithms themselves.
2) Once the base classifiers are built, their decisions on the training instances are also gathered. For supervised feature selections methods, a class label is required for each training sample, the same class attribute is taken from the original data set, and assigned to each of the instances. A new dataset is therefore constructed, each column represents an artificially generated feature, each row corresponds to a training instance, the cell then stores the transformed feature value.
3) A new feature selection algorithm "Feature Selection with Harmony Search"(HSFS) [29] 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 sizes. When harmony search algorithm terminates, its best harmony is translated into a feature subset and returned as the feature selection result.
4) Once the classifier ensemble is constructed, new objects are classified by the ensemble members, and their results are aggregated to form the final ensemble decision output.

In the training part, this module will not give any output, but only receives $D_{training}(\overrightarrow{P}, T_{stroke})$ from the Gesture Sampling Module. However, in the working part, the module receives $D_{working}(\overrightarrow{P})$ and the gives its prediction result $R(stroke)$ to the Stroke Generation module.

C. Stroke Generation

The Stroke Generation module is response to use the identified gestures to produce their stroke trajectories. The trajectories of the five elementary strokes are designed and implemented before the module starts to work. The vertical, the horizontal and the fold strokes are implemented straight line. However, the left falling stroke and the right falling stroke requires a certain level of radian. Therefore, the parabola fitting algorithm is applied to generate such two trajectories. Besides the stroke type, the parabola fitting algorithm requires extra two positions to locate each stroke's position in the canvas. The the start point position of a stroke $P_s(x_{start}, y_{start})$ and the end point position $P_e(x_{end}, y_{end})$ are used for the two positions.

Note that: the human gesture trajectories $\overrightarrow{P}$ cannot be used as $P_s(x_{start}, y_{start})$ or $P_e(x_{end}, y_{end})$ directly. We need to apply the following equations to convert:

$$\begin{cases} x_{p_{s/e}} = 20 \cdot \dfrac{x_{\overrightarrow{P}} - \frac{H}{2}}{H} \\ y_{p_{s/e}} = 20 \cdot \dfrac{y_{\overrightarrow{P}} + H}{H} \end{cases} \tag{4}$$

where: $x_{\overrightarrow{P}}$ and $y_{\overrightarrow{P}}$ are the human gesture trajectory's data; $x_{p_{s/e}}$ and $y_{p_{s/e}}$ are the stroke position in the canvas; s/e indicates the position is belonged to the start point or the end point; H is a scale parameter that is determined by the robotic arm's configuration; H is set to 480 in this paper.

The stroke generation method of the five strokes are introduced as follows: Set $P_s(x_{start}, y_{start})$ as the start point position of a stroke, and set $P_e(x_{end}, y_{end})$ as the end point position,

1) Horizontal Stroke: move the pen from the start point P_s to the end point P_e directly.
2) Vertical Stroke: the same procedure with the horizontal stroke, move the pen from the start point P_s to the end point P_e directly.
3) Fold Stroke: this stroke is divided into components. The robot arm use the horizontal stroke pattern before the arm moves the inflection point P_i of the fold stroke, and then, the arm switches to the vertical stroke pattern. The x value of the end point P_e and the y value of start point P_s compose the inflecting point position, thus, the inflecting point is $P_i(x_{end}, y_{start})$. Therefore, the entire trajectory is from P_s via P_i to P_e
4) Left Falling Stroke: The parabola fitting algorithm is applied to generate the trajectory of the left falling stroke. The equation of the parabola fitting is:

$$y = ax^2 + bx + c, \quad x \in \left[x_{start}, x_{end}\right] \tag{5}$$

where: a, b and c are the three parameters defining a parabola's shape. Merely P_s and P_e are not enough for calculating a, b and c; thus, P_i in the fold stroke is required for the calculation.

$$\begin{cases} a = \frac{y_{start} - y_{end}}{(x_{start} - x_{end})^2} \\ b = \frac{-2x_{end}(y_{start} - y_{end})}{(x_{start} - x_{end})^2} \\ c = y_{end} + \frac{b^2}{4a} \end{cases} \tag{6}$$

Then, 8 points between P_s and P_e are selected to fit the stroke trajectory. The 8 points are represented as $(x_1, y_1), (x_2, y_2), \cdots, (x_8, y_8)$. Equation 7 defines the x values of the 8 points.

$$x_k = x_{start} + k\frac{x_{end} - x_{start}}{10} \tag{7}$$

Then, bring x_k to Equations 5 to calculate y_k.
5) Right Falling Stroke: The entire calculation procedure of this stroke is identical with the left falling stroke's.

After the parabola fitting process, each human gesture trajectory $D_{training}(\overrightarrow{P})$ has been converted to an array of points $\overrightarrow{s}(x_k, y_k), k \in [1, 9]$. The array is sent to the Robotic Arm Control module, 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.

D. Robotic Arm Control

The robotic arm cannot access the stroke trajectories directly. The arm only executes joint value commands. In this case, the reverse kinematic algorithm is required to transform the stroke trajectories to the arm's joint values.

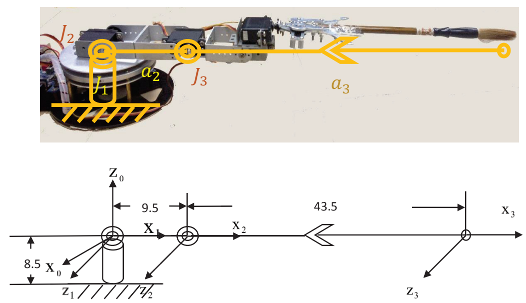

Fig. 5. The configuration of the robotic arm

The top picture in Figure 5 shows the experimental robotic arm setup. Three joints, which are labeled as J_1, J_2, and J_3, are used in the work. In addition, a_2 presents the distance between the second joint and the third joint; a_3 indicates the distance between the third joint and the top of the brush.

The bottom picture in Figure 5 illustrates the coordinate systems of the three joints and the mechanical parameters. By using D-H method and the inverse kinematics formula, we obtain the following equations to calculated θ_1, θ_2 and θ_3. Note that z is is the distance between the brush and the shoulder, x_k and y_k are the trajectory's x, y components.

$$z = \frac{a_2 + a_3 - \sqrt{a_2^2 + a_3^2}}{2} + \sqrt{a_2^2 + a_3^2} \tag{8}$$

$$\theta_1 = \arctan\frac{z}{x_k} \tag{9}$$

$$\theta_3 = \arccos\frac{(z\cos\theta_1 + x_k \sin\theta_1)^2 + y_k^2 - a_2^2 - a_3^3}{2a_2a_3} \tag{10}$$

$$\varrho = a_3 \sin\theta_3 \sqrt{a_2^2 + a_3^2 + 2a_2a_3\cos\theta_3 - y_k^2} \tag{11}$$

$$\theta_2 = \arcsin\frac{a_1 y_k + a_3\cos\theta_3 y_k + \varrho}{a_2^2 + a_3^2 + 2a_2a_3\cos\theta_3} \tag{12}$$

where: a_2 is $9.5mm$ and a_3 is $43.5mm$. Thus, we are able to obtain the values of θ_1, θ_2 and θ_3 by using Equations. 9, 12 and 10.

E. The Experimental System

Figure 6 shows the experimental system. The system contains a robotic arm with 6 DOFs. The arm is mounted on a workspace, and 3 DOFs of the arm are used to perform writing movements. A brush pen is mounted on the top of the arm. This setup allows the robotic arm to have enough DOFs to act in a 3-dimensional environment. Each rotational joint of the robot arm has a motor driver and also an encoder

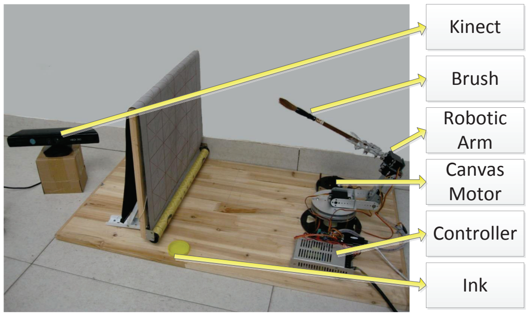

Fig. 6. The experimental system

that senses the motor's position. A writing board facing the arm is fixed vertically. A canvas is installed on the board. The canvas will turn to black if water touches the canvas, and the black area will disappear when the canvas is dry. A canvas motor drags the canvas when the robot completes one character, so that the robots can always write a new character in the blank area of the canvas.

Both the robot and the canvas motor are controlled by the hardware controller AVR computer, which is placed near the arm. A small bottle of water is used as ink. The arm puts the brush into the bottle when the human demonstrator sends a command. A Kinect device is set up separately to face the human demonstrator. In our experiments, the Kinect's sampling rate is about 30 frames per second, and its image output resolution is 320×240.

IV. Experimentations

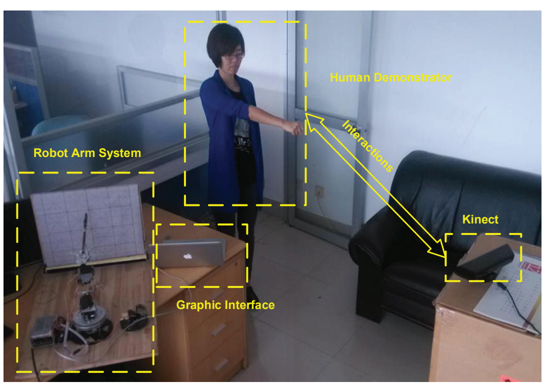

Fig. 7. The experimental setup

Figure 7 illustrates the positions of the human demonstrator, the user interface, the Kinect device and the robotic arm system. The demonstrator stands within the detection range of the Kinect. A laptop computer running the graphic user interface is placed facing to the demonstrator, so that, the demonstrator is able to adjust his gestures. The robotic arm system is placed separately. For the software implementation, an algorithm computer is for the gesture sampling, the classifier ensemble reduction, and the stroke generation modules. The robotic arm is driven by an AVR computer. The programs in the algorithm computer are developed by using the C# programming language and the "Microsoft Kinect SDK 1.5"; the robotic control program is written in C++.

TABLE I
Number of Sampling Gestures

Strokes	Horizontal	Vertical	Left Falling	Right Falling	Fold
Instances	220	217	208	219	206

The experimental procedure is also divided into two parts: 1) Classifier ensemble training part, and 2) Human gesture guided robotic writing part. In Part 1, five persons join the experiment to perform the five predefined gestures. Each person performs about 40 times for each gesture. The captured dataset are used to train the classifier ensemble. In Part 2, only one human demonstrator stands in front of the Kinect device to demonstrate a Chinese character with his/her right arm. the demonstrator adjusts his/her gestures via the interaction interface window. Once the arm receives the stroke's trajectories, the robotic arm starts to move. Therefore, the arm's writing actions have a very short delay. Usually, when the demonstrator finishes demonstrating a character's strokes, the robot has almost completed the writing.

A. *Reduction Performance for Gesture Classification*

Table I lists the number of the samples of the five elementary gestures. In order to extend the classifier's generalization, the gestures are performed by 5 different persons. Each category consists of more than 200 samples. Hence, the entire training dataset contains more then 1000 samples in total.

To demonstrate the capability of the proposed CER framework, a number of experiments have been carried out. 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 II summarizes the obtained three sets of results for CFS. After applying CER, as compared against the results of using: 1) the base algorithm itself, 2) the full base classifier pool, and 3) randomly formed ensembles. Several general observations can be drawn across all three setups. First of all, the prediction accuracies of the constructed classifier ensembles are universally superior than that achievable by a single C4.5 classifier. The accurate rates of the three types of C4.5 are less than 90%. Most of the datasets that revealed the most performance increase are either large in size or high in dimension. This confirms the benefit of employing classifier ensembles. All feature selection techniques tested demonstrate substantial ensemble size reduction, showing clear evidence of dimensionality reduction. Based on the observation of the table, in order to use the smallest ensemble size to achieve relatively good performance, the ensemble size is set as 30. After the testing the classifier ensemble's performance, the experiment switches to Part 2.

TABLE II
C4.5 BASED ENSEMBLE CLASSIFICATION ACCURACY RESULT COMPARISON

	CFS		Random		Full		
Pool Size	Acc.	Size	Acc.	Size	Acc.	Size	C4.5
50	91.40	22.2	89.53	10	**92.06**	50	87.38
100	**91.50**	28.2	90.84	20	91.21	100	86.73
200	**91.31**	30.4	91.21	40	91.04	200	87.20

B. Performance of Robotic Writing

Figure 8 demonstrates the robot's writing results of the predefined five strokes. To easily compare the writing quality, the print versions are listed in the left column. The writing result of the horizontal and the vertical stroke are satisfactory. However, the two trajectories are not straight enough. We believe the situation is caused by the limitations of the robotic arm, which cannot give very high repeated positioning accuracy. The left falling stroke's quality is excellent; the its entire trajectory is smooth, and the shape is very close to the print version's. The results of the right falling stroke and the fold stroke are not good enough. In particular, there exists a gap between the writing result and the print version: the robot arm does not execute a horizontal drawing when the arm writes the tail part of the trajectory. This gap might be solved by improving the trajectory fitting algorithm. As a summary, the system's classifier successes to recognize the human demonstrator's gesture, and the stroke pattern is able to generate accurate stroke trajectories.

Figure 9 shows the writing results of three simple Chinese characters. The first and the third characters contains three strokes, and the second has four in total. The characters in the left column of the figure are the printed style of the Chinese characters. The characters in the right column are generated by the robot. This figure shows that the Kinect device successfully captures the human action sequence, since these three characters consist of simple structures and slightly-changed strokes. For the those sequential gesture trajectories, the classifier ensemble algorithm also produces high recognition accuracy, each stroke's type of the characters is precise. In addition, the layouts of the strokes are exactly correct; especially, in the third character, the starting and the end positions of each stroke are almost in the same position. This indicates that the approach's trajectory conversion algorithm works properly. Also, compared with characters' printing style, the effects of the robot's writing are good.

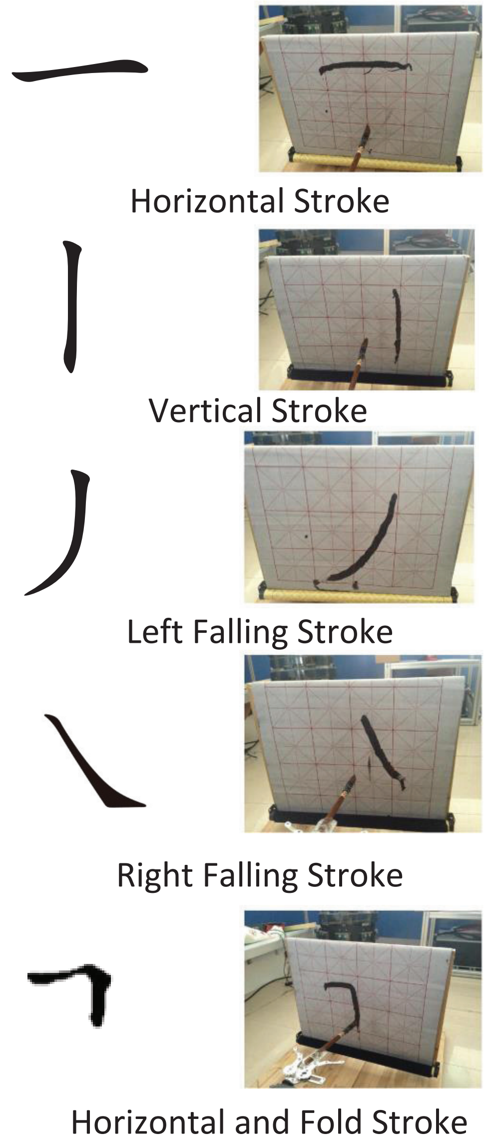

Fig. 8. The robotic writing results of the typical strokes

V. CONCLUSIONS

This paper has presented a human gesture guided approach to robotic Chinese character writing. The approach is achieved by (1) using a Kinect device to obtain human gesture's information; (2) applying classifier ensemble algorithms to recognize captured trajectories; (3) employing feature selection with harmony search method to optimize the recognition accuracy; (4) invoking parabola fitting algorithm to generate stroke trajectories; and (5) using inverse kinematic algorithm to obtain robotic joint parameters. The observations from the experiments demonstrate that using human gesture can conveniently transform Chinese character font information to the robot arm; the classifier ensemble is able to recognize the human gestures with high accuracy; and the robot system is able to easily write many simple Chinese characters without complex calculations and image processing. In addition, the writing quality is good, especially, each stroke's position is exactly correct.

In the present work, we only use human arm gestures to obtain character's information. In our future work, we propose to send more information of a human's gestures to the robot system, such as wrist orientation; thereby increasing our robotic ability to write characters that are more

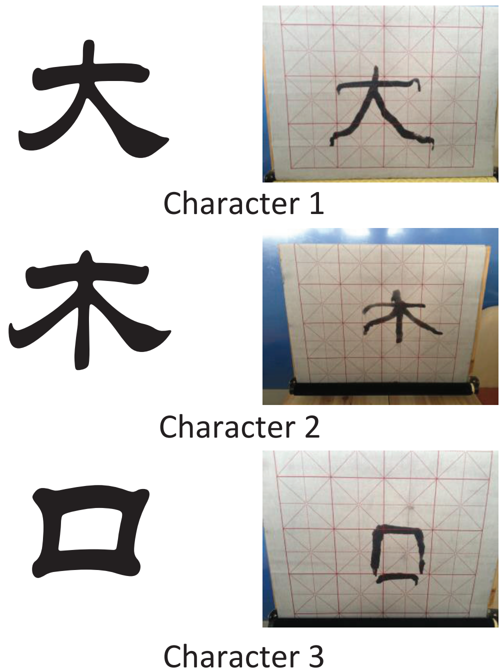

Fig. 9. The writing results

human-like. Also, the robot only follows optimized stroke trajectories to write. In our future work, we will focus on building robotic learning algorithms that allows the robot to gradually learn better writing skill through autonomous attempt movements and human-robot interactions. Furthermore, the current stroke trajectories are generated by the fitting algorithm, we will also use human gesture to guide the robot to produce the stroke trajectories.

References

[1] N. Huebel, E. Mueggler, M. Waibel, and R. D'Andrea, "Towards robotic calligraphy," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vilamoura: IEEE Press, 2012, pp. 5165–5166.

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

[3] S. Arimoto, H. Hashiguchi, and R. Ozawa, "A simple control method coping with a kinematically ill-posed inverse problem of redundant robots: analysis in case of a handwriting robot," *Asian Journal of Control*, vol. 7, no. 2, pp. 112–123, 2005.

[4] F. Chersi, "Learning through imitation: a biological approach to robotics," *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 3, pp. 204–214, 2012.

[5] J. P. Bandera, J. A. Rodríguez, L. Molina-Tanco, and A. Bandera, "A survey of vision-based architectures for robot learning by imitation," *International Journal of Humanoid Robotics*, vol. 9, no. 1, pp. 1 250 006–1–1 250 006–40, 2012.

[6] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[7] G. Tsoumakas, I. Partalas, and 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.

[8] H. Zhou, H. Huo, H. Liu, and J. Tang, "Classification of upper limb motion trajectories using shape features," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 970–982, 2012.

[9] H. I. Suk, B. K. Sin, and S. W. Lee, "Hand gesture recognition based on dynamic bayesian network framework," *Pattern Recognition*, vol. 43, no. 9, pp. 3059–3072, 2010.

[10] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on System, Man, and Cybernetics, Part C*, vol. 37, no. 3, pp. 311–324, 2007.

[11] J. Richarz and G. A. Fink, "Visual recognition of 3d emblematic gestures in an hmm framework," *Journal of Ambient Intelligence and Smart Environments*, vol. 3, pp. 193–211, 2011.

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

[13] R. Diao, F. Chao, T. Peng, N. Snooke, and Q. Shen, "Feature selection inspired classifier ensemble reduction," *IEEE Transactions on Cybernetics*, In press.

[14] G. Giacinto and F. Roli, "An approach to the automatic design of multiple classifier systems," *Pattern Recognition Letter*, vol. 22, no. 1, pp. 25–33, 2001.

[15] I. Partalas, G. Tsoumakas, and I. Vlahavas, "Pruning an ensemble of classifiers via reinforcement learning," *Neurocomputing*, vol. 72, no. 7–9, pp. 1900–1909, 2009.

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

[17] X. Ma, Q. Kong, W. Ma, and X. Zhang, "4-dof lettering robot's trajectory planning," *Mechanical Engineering and Automation*, vol. 165, no. 5, pp. 161–163, 2010.

[18] G. Wang, Z. Liao, and X. Chen, "Robot manipulator lettering technology and motion," *Journal of Chongqing University*, vol. 26, no. 12, pp. 6–9, 2003.

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

[20] J. H. M. Lam and Y. Yam, "Application of brush footprint geometric model for realization of robotic chinese calligraphy," in *IEEE International Conference on Cognitive Infocommunications*. Budapest: IEEE Press, 2011, pp. 1–5.

[21] S. Yussof, A. Anuar, and K. Femandez, "Algorithm for robot writing using character segmentation," in *IEEE International Conference on Information Technology and Applications*. Sydney: IEEE Press, 2005, pp. 21–24.

[22] K. W. Lo, K. W. Kwok, S. M. Wong, and Y. Yam, "Brush footprint acquisition and preliminary analysis for chinese calligraphy using a robot drawing platform," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Beijing: IEEE Press, 2006, pp. 5183–5188.

[23] K. W. Kwok, K. W. Lo, S. M. Wong, and Y. Yam, "Evolutionary replication of calligraphic characters by a robot drawing platform using experimentally acquired brush footprint," in *IEEE International Conference on Automation Science and Engineering*. Shanghai: IEEE Press, 2006, pp. 466–471.

[24] F. Yao, G. Shao, and J. Yi, "Extracting the trajectory of writing brush in chinese character calligraphy," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 6, pp. 631–644, 2004.

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

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

[27] D. Pérez-Marcos, J. A. Buitrago, and F. D. G. Velásquez, "Writing through a robot: A proof of concept for a brain-machine interface," *Medical Engineering and Physics*, vol. 33, no. 10, pp. 1314–1317, 2011.

[28] Y. Sun and Y. Xu, "A calligraphy robot - Callibot: design, analysis and applications," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*. IEEE, 2013, pp. 185–190.

[29] R. Diao and Q. Shen, "Feature selection with harmony search," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 6, pp. 1509–1523, 2012.