Visual Gesture Character String Recognition by Classification-Based Segmentation with Stroke Deletion

Xiao-Jie Jin, Qiu-Feng Wang, Xinwen Hou, Cheng-Lin Liu
National Laboratory of Pattern Recognition (NLPR)
Institution of Automation of Chinese Academy of Sciences
P.O. Box 2728, Beijing 100190, P.R. China
{xiaojie.jin, wangqf, xwhou, liucl}@nlpr.ia.ac.cn

***Abstract*—The recognition of character strings in visual gestures has many potential applications, yet the segmentation of characters is a great challenge since the pen lift information is not available. In this paper, we propose a visual gesture character string recognition method using the classification-based segmentation strategy. In addition to the character classifier and character geometry models used for evaluating candidate segmentation-recognition paths, we introduce deletion geometry models for deleting stroke segments that are likely to be ligatures. To perform experiments, we built a Kinect-based fingertip trajectory capturing system to collect gesture string data. Experiments of digit string recognition show that the deletion geometry models improve the string recognition accuracy significantly. The string-level correct rate is over 80%.**

Keywords—Visual gesture character string; string recognition; over-segmentation; deletion geometry model; Kinect

I. INTRODUCTION

Online and offline handwriting recognition, including handwritten character recognition and string recognition, has been studied for nearly a half century and many effective methods have been proposed. This technology finds a lot of applications where handwriting is generated by pen-based devices or paper document scanning. In such applications, the characters are normally separated by pen lifts through the differentiation between between-character gap and within-character gap. A string of characters is usually recognized by integrating character segmentation and recognition, i.e. using a character classifier and context models to evaluate the paths of candidate segmentation-recognition combinations and the optimal path gives the character segmentation and class labels simultaneously.

In recent years, the combination of handwriting recognition with computer vision systems has attracted interests. A protocol is to capture handwriting stroke trajectory using a video camera [1-5]. Such technology is very useful in human-machine interaction (including games, TV and education systems) to transmit rich semantic information to the machine. Unlike pen captured or written-on-paper handwriting, the visual character strings (also called gesture strings) have no pen lift information, i.e. a string is a single stroke. Some examples of gesture strings collected by our Kinect-based fingertip trajectory capturing system [14] are shown in Fig. 1. The difficulty of character segmentation without pen lifts poses a great challenge to the recognition of gesture character strings.

Researchers have paid many efforts for handwriting recognition in computer vision system and introduced some creative systems [1-5], which can be divided into two categories. In one category [1, 2, 3], handwriting was produced with an ordinary pen on paper, thus the ink trace recorded by a camera could be used to detect the pen-up state. Munich et al. [1] presented a camera-based human-computer interface to acquire handwriting, where statistical models were used to classify the strokes as either pen-up or pen-down state. Based on the method of [1], Fink [2] reported handwriting recognition results using the information of video input. Bunke et al. [3] proposed to extract the ink trace using differential images and presented the results on a small writer-dependent data set. In the other category, the characters were written by moving a finger on the desk or in the air. The finger trajectory was recorded by a camera or other motion sensors, e.g., Kinect [4, 5]. These systems are more intuitive and convenient for users to input characters than the written-on-paper systems. However, it complicates the segmentation of characters due to the unavailability of pen lift in the moving trajectory, and so, these systems can only recognize isolated characters. Long and Jin [4] proposed a hybrid system for finger-writing characters. Feng et al. [5] proposed a Kinect-based system to track and recognize written-in-the-air characters.

The above works either used the pen lift information to help segment characters or just recognized isolated characters. To our best of knowledge, there is no reported work on vision-based gesture character string recognition, where there is no pen lift information.

This paper reports our first attempt to gesture character string recognition. We use a classification-based segmentation strategy with character over-segmentation and deletion of extra strokes (ligatures, see Fig. 1). To detect and delete ligatures, we propose a deletion geometric model to integrate in the segmentation-and-recognition framework with the other contexts. We evaluated the performance of the proposed method on 830 gesture digit strings collected by our Kinect-based fingertip trajectory capturing system, and obtained high recognition accuracy.

The rest of this paper is organized as follows. In Section 2, we outline the system structure and the processing steps; Section 3 describes the path evaluation criterion, including the deletion geometric model; Section 4 presents our experimental results and Section 5 offers concluding remarks.

978-1-4799-2190-4/13 $26.00 © 2013 IEEE
DOI 10.1109/ACPR.2013.7

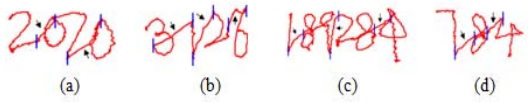

Fig. 1. Four examples of gesture digit strings. Strokes between the short vertical line and indicated by arrows are extra strokes (ligatures).

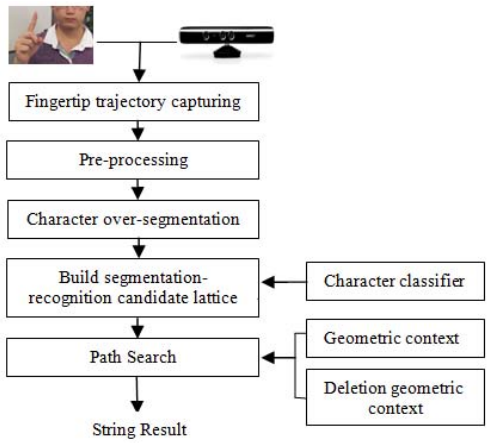

Fig. 2. Block diagram of the recognition system.

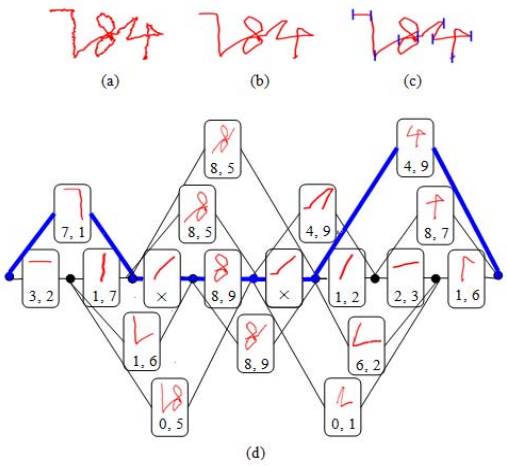

Fig. 3. (a) Original gesture string. (b) Smoothed gesture string. (c) Over-segmentation of (b), the short vertical lines indicate segmentation points. (d) Segmentation-recognition candidate lattice of (c), the upper part of each box is the candidate character pattern, and the lower part shows its top two candidate character classes. Symbol "×" denotes a stroke to be deleted. The optimal path is labeled by the thick blue line.

II. System Overview

To obtain gesture string data, we built a Kinect-based fingertip trajectory capturing system. Since it is difficult to segment characters correctly without pen lift information in gesture strings, we apply classification-based segmentation strategy based on character over-segmentation with deletion geometric context. Fig. 2 shows the block diagram of our complete system, which comprises five main processing steps as follows.

1) Fingertip trajectory capturing: To simplify this procedure, we require the user writes with his/her writing finger pointing upward during writing. Firstly, we extract user's body from the environment background. Secondly, we get the skeletal node coordinate of the writing hand from skeletal tracking data. By coordinate transformation, the node's coordinate in the depth image is obtained. Thirdly, we search in the depth image around the writing hand's skeletal node position to find an area that only contains the hand. Fourthly, we localize the fingertip by simply searching the area got in the last step in the order from top to bottom and from left to right. The first point that belongs to the hand is thought to be the fingertip. By connecting detected fingertip in every frame, we obtain the fingertip trajectory, which gives a gesture string.

2) Preprocessing: The trajectory of gesture string is smoothed by replacing each sample point with the mean value of its neighbors. Redundant points are removed to ensure that only one point left in the same position in the trajectory (Fig. 3b).

3) Character over-segmentation: Over-segmentation points are detected to split the string into stroke segments such that each segment is a character, a part of character or an extra stroke (Fig. 3c). Over-segmentation points are sample points with following properties: high maximum of local curvature, low minimum of writing velocity, end points of single-stroke regions in the x-coordinate projection of trajectory.

4) Build segmentation-recognition cadidate lattice: One or more consecutive stroke segments are concatenated to generate candidate character patterns, and each candidate pattern is classified to assign candidate character classes with corresponding classification scores or labeled as ligature, forming a segmentation-recognition lattice (Fig. 3d). The corresponding classification scores are stored to be fused with contexts.

5) Path search: Each character class (or ligature) sequence paired with candidate character patterns (this pair is called a candidate segmentation-recognition path) is evaluated by fusing multiple contexts (character classification, character geometric context, deletion geometric context), and the optimal path is searched via dynamic programming (DP) to give the segmentation and recognition result.

In the above, we can see that the string recognition problem is formulated as a problem of path evaluation and search in the segmentation-recognition candidate lattice. While the DP search algorithm is off-the-shelf (e.g., in [6]), the evaluation of paths is radical to the recognition performance.

III. Path Evaluation

It is important to design a path evaluation criterion to assign the highest score to the truth segmentation-recognition path while assigning lower scores to the other paths. Usually, a path is evaluated by combining character classification score, linguistic context and character geometric context [6, 7]. For the recognition of digit strings in our experiment, there is no linguistic context constrain. Moreover, the gesture string may include many extra strokes, which should be deleted during recognition. Hence, we design deletion geometric models

considering unary and binary deletion geometric contexts and integrate them in the path evaluation criterion with character classification score and character geometric context model.

In the following, we first describe our path evaluation criterion. Next, we introduce the context models used in the criterion. Lastly, we present the combining weights learning method.

A. Path evaluation criterion

In the segmentation-recognition lattice (see Fig. 3d), each path can be represented by a pair of candidate character pattern sequence ($O=o_1,\cdots,o_n$) and candidate character class sequence ($C=c_1,\cdots,c_n$), where the number n represents the length of this path (character number). Since there may be extra strokes, each candidate character class c_i is either from a valid character class set by a character classifier ($c_i \in \{\omega_1,\cdots,\omega_M\}$, M is the number of character classes) or an invalid class ($c_i=\omega_0$, i.e. extra stroke). The optimal segmentation-recognition result is decided by the path with the highest score:

$$(O^*,C^*)=\arg\max_{(O,C)} f(O,C), \tag{1}$$

where $f(O,C)$ is a combination of character classification score (**cls**), character geometric context and deletion geometric context. We denote the set of extra strokes that are deleted in a path as *D*, and the set of normal patterns (un-deleted) as *N*. The path score $f(O,C)$ is then:

$$f(O,C)=\sum_{o_i\in N}[k_i \log P(c_i\mid o_i)+\lambda_1 \log P(c_i\mid g_i^{cuc})+\lambda_2 \log P(z_i^p=1\mid g_i^{cui})]$$
$$+\sum_{o_i\in D}\{\lambda_3 \log P(\omega_0\mid g_i^{du})+\lambda_4[P(\omega_0\mid o_{i-1}\in N, g_{i,i-1}^{db})+P(\omega_0\mid o_{i+1}\in N, g_{i,i+1}^{db})]\}, \tag{2}$$

where $P(c_i\mid o_i)$, $P(c_i\mid g_i^{cuc})$ and $P(z_i^p=1\mid g_i^{cui})$ are the scores of character classification, character unary class-dependent geometric model (**cucg**) and character unary class-independent geometric model (**cuig**) (similar to those in [7]), respectively. $P(\omega_0\mid g_i^{du})$ is the score of deletion unary geometric model (**dug**), which measures the likeliness of stroke deletion. $P(\omega_0\mid o_{i+1}\in N, g_{i,i+1}^{db})$ and $P(\omega_0\mid o_{i-1}\in N, g_{i,i-1}^{db})$ are binary geometric models (**dbg**), which score the binary geometric features between the deleted stroke o_i and its neighboring character patterns o_{i-1}, o_{i+1}. Inspired by the variable length HMM of [11], we use the number of primitive segments forming a candidate character pattern k_i to overcome the bias of path score to short strings. The terms $\{\lambda_1,\lambda_2,\lambda_3,\lambda_4\}$ are four weights to balance different models' effects. The summation nature of path score $f(O,C)$ in Eq. (2) guarantees that the optimal path can be found by DP search.

B. Deletion geometric models

In the following, we consider both unary and binary geometric contexts for modeling deleted strokes.

1) Deletion unary geometric model

By observing the extra strokes that should be deleted in gesture strings (see Fig. 1), we notice that they usually have distinct geometric features from other valid character patterns, e.g. the geodesic distance of them is short, the direction from the stroke's start point to its end point is usually from left to right and the stroke complexity (defined as the geodesic distance divided by the straight length between the stroke's two end points) is usually small. Accordingly, as shown in Table I, we extract three features to represent the deletion unary geometric context. Such features indicate whether a stroke should be deleted or not. For this two-class problem, we use a linear support vector machine (SVM) trained with deleted strokes and normal stroke samples.

2) Deletion binary geometric model

The deletion unary geometric context is not robust enough, because in this case some strokes which ought to be part of characters might be wrongly deleted, as shown in Fig. 4. Thus, we also exploit the geometric features between the deleted stroke and its neighboring character patterns to help evaluate the likelihood of deletion. As a part of its neighboring characters, the stroke usually overlap with the neighboring characters. Table II lists the nine features that we extract for deletion binary geometric context. Among them, No. 1-4 measures the relationship between the deleted stroke and its neighboring character patterns, and No. 5-9 characterize the bounding box after they are merged. The calculation of horizontal/vertical overlap is referred to [9].

Similar to the deletion unary geometric model, we use a two-class linear SVM to score these features to help discriminate deleted and un-deleted strokes.

Fig. 4. Error examples when only "**dug**" is used. The strokes indicated by arrows are treated to be deleted strokes. However, they are parts of digit "2" and "9", respectively. These stokes can be retained after "**dbg**" is used.

C. Character Geometric models

Based on the work of [8], we extract nine features for "**cucg**" and "**cuig**", which are listed in Table III. All of these features are related to the bounding box of the character. On obtaining such character geometric features, we use a quadratic discriminant function (QDF) to model the class-dependent features, and for the class-independent features, we similarly use a linear SVM.

D. Confidence transformation

In order to convert the output of character classifier, the geometric models "**dug**", "**dbg**", "**cucg**" and "**cuig**" into posterior probabilities, we apply sigmoidal confidence transformation for each one of them. The sigmoidal function is defined as

$$p(\omega_j\mid x)=\frac{\exp[-\alpha d_j(x)+\beta]}{1+\exp[-\alpha d_j(x)+\beta]}, j=1,\cdots,M \tag{3}$$

where M is the number of defined classes, $d_j(x)$ is the dissimilarity score of class ω_j, α and β are the confidence parameters. We optimize α and β by minimizing the cross entropy (CE) loss function [9] on a validation data set.

TABLE I. DELETION UNARY GEOMETRIC FEATRURES.

No.	Feature	Norm[a]
1	Geodesic length of stroke	Y
2	Angel between the line from the stroke's start point to its end point and the horizontal line	N
3	Stroke complexity	N

[a.] Denotes whether normalized w.r.t the gesture string height or not

TABLE II. DELETION BINARY GEOMETRIC FEATURES.

No	Feature	Norm
1-2	Horizontal / vertical overlap	N
3-4	Distance between horizontal / vertical geometric center	Y
5-6	Width / height of bounding box	Y
7	Diagonal length of bounding box	Y
8	Square root of bounding box	Y
9	Logarithm of aspect ratio	Y

TABLE III. CHARACTER UNARY CLASS-DEPENDENT / CLASS-INDEPENDENT GEOMETRIC FEATRUES

No	Feature	Norm
1-2	Width / height of bounding box	N
3-4	X / Y coordinates of the geometric center of bounding box	Y
5-6	Distance from the upper / lower bound to the string horizontal center line	Y
7	Diagonal length of bounding box	Y
8	Square root of bounding box	Y
9	Logarithm of aspect ratio	Y

E. Optimization of combining weights

The combining weights $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$ include character geometric models related weights $\{\lambda_1, \lambda_2\}$ and deletion geometric models related weights $\{\lambda_3, \lambda_4\}$. To adjust these weights, we adopt a two-step procedure.

1) Firstly, we pick up training string samples without deleted strokes, i.e. samples that can still be recognized correctly without using deletion geometric models. In this case, the deletion geometric models related weights $\{\lambda_3, \lambda_4\}$ in Eq. (2) are thought to be zero, thus we could easily adjust the character geometric models related weights $\{\lambda_1, \lambda_2\}$ to achieve the maximum accuracy on these samples.

2) Secondly, we fix the character geometric models related weights $\{\lambda_1, \lambda_2\}$, and then we adjust $\{\lambda_3, \lambda_4\}$ on the training samples with deleted strokes to achieve the maximum accuracy on training samples.

IV. EXPERIMENTS

We evaluate the performance of our approach on a gesture digit string data set collected using Kinect. The experiments were performed on a desktop computer of 2.66GHz CPU, programming using Microsoft Visual C++. In the step of fingertip trajectory capturing, the resolution of depth image was set to be 320×240, and fingertip was tracked at 30 fps.

A. Experimental setting

By our Kinect-based fingertip trajectory capturing system, we collected 830 gesture digit strings written by 30 persons. In this data set, ten digits appear at roughly the same frequency, and the length of digit strings varies from 2 to 6, with most 4-digit strings (67 percent).

We divided the strings into a training set of 10 writers and a test set of 20 writers. The training set contains 200 strings for training the geometric models ("**dug**", "**dbg**", "**cucg**", "**cuig**"), confidence parameters and the combination weights, and the test set contains 630 strings for evaluating the string recognition performance.

For digit classification, we choose a Nearest Prototype Classifier (NPC) classifier. Each character pattern is represented by a 512-dimensional feature vector using a normalization-cooperated method for 8-direction feature extraction [12]. The feature vector is further reduced to 200D by Principle Component Analysis (PCA) for computational efficiency. Our classifier was trained with the digit samples extracted from the CASIA-OLHWDB database [15], which contains 10,141 digits, 8,000 of which are used for training and the rest are used as validation data set. To simulate visual gesture characters without pen lift, all the pen lifts in the training digit samples were filled with straight lines to connect the adjacent strokes.

The performance is evaluated using the string-level accuracy metric: String-level Correct Rate (SCR):

$$\text{SCR} = N_c / N_t \qquad (4)$$

where N_c is the number of strings correctly recognized, while N_t is the number of total test strings. A test string is recognized correctly only when the result string is exactly the same as the ground-truth (transcript).

B. Experimental results

The string recognition accuracies by combining different contexts are shown in Table IV. We can see that the accuracy is very low when only character classifier ("**cls**") is used (case 1), and it is improved to the SCR of 17.7% by adding character geometric models (case 2). On the other hand, the unary and binary deletion geometric models improve the SCR significantly to 58.9% (case 3) and 28.5% (case 4), respectively. And the combination of them improves the SCR to 62.7% (case 5). This demonstrates the effectiveness of deletion geometric models, especially the deletion unary geometric model. Furthermore, by combining the character geometric models with the deletion geometric models, the SCR is improved further in all cases (case 6-8). This justifies that the deletion geometric context and character geometric context are complementary. The best result, SCR of 81.6%, is obtained by combining all the context models.

The string recognition results on strings of different lengths are listed in Table V, from which we can see that the SCR is very high on short strings (i.e. strings of length 2, 3) but decreases when the string length increases. This is because increased string length increases the difficulty of character segmentation and the probability of mis-segmentation and mis-classification.

TABLE IV. PERFORMANCE OF THE COMBINATION OF DIFFERENT CONTEXTS

cases	cls	cucg+cuig	dug	dbg	SCR (%)
1	✓				8.8
2	✓	✓			17.7
3	✓		✓		58.9
4	✓			✓	28.5
5	✓		✓	✓	62.7
6	✓	✓	✓		78.3
7	✓	✓		✓	50.1
8	✓	✓	✓	✓	81.6

TABLE V. PERFORMANCE ON STRINGS WITH DIFFERENT LENGTH

String length	2	3	4	5	6
Percentage(%)	11	12	67	6	4
SCR (%)	96.3	86.5	80.5	71.7	61.9

C. Error analysis

By analyzing the experimental results, we summarize the string recognition errors into three categories: over-segmentation error, classification error and deletion error. In over-segmentation error (one example in Fig. 5a), the characters are not correctly separated (a character is correctly over-segmented when it is separated from other characters despite the within-character splits), then these characters cannot be correctly recognized by combining consecutive primitive segments. Character classification error (one example in Fig. 5b) implies that the correct class of the candidate pattern is not in the best path by the path search method. By the deletion error (one example in Fig. 5c), some valid strokes are mis-classified to deletion strokes, and then they are deleted.

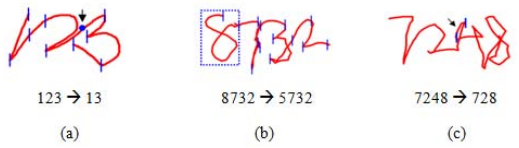

Fig. 5. Three examples of recognition errors. (a) Over-segmentation error, the segmented point indicated by the arrow is missed; (b) Classification error, the first character is mis-classified to "5"; (c) Deletion error, the valid stroke indicated by the arrow is treated as a deletion stroke.

V. CONCLUSIONS

We propose a visual gesture character string recognition method using classification-based character segmentation with stroke deletion. We introduce deletion geometric models for deleting strokes that are likely to be ligatures and integrate them with character classifier and character geometric contexts in the integrated segmentation-recognition framework. In experiments on digit strings captured using Kinect, we have achieved fairly high string recognition accuracies. Particularly, the deletion geometric models can improve the string recognition accuracy significantly. In our future work, we will improve the deletion geometric models by extracting more discriminative features, and extend the character set to include English letters and Chinese characters.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) Grant 60933010 and the Key Deployment Program of Chinese Academy of Sciences Grant KGZD-EW-103-5(8).

REFERENCES

[1] M. E. Munich, P. Perona, Visual Input for Pen-Based Computers, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 313-328, Mar, 2002

[2] G. A. Fink, M. Wienecke, G. Sagerer, Video-Based On-line Handwriting Recognition, *Proc.6th Int'l Conf. Document Analysis and Recognition*, Seattle, WA, pp. 226-230, 2001

[3] H. Bunke, T. von Siebenthal, T. Yamasaki, M. Schenkel, Online handwriting data acquisition using a video camera, *Proc.5th Int'l Conf. Document Analysis and Recognition*, Bangalore, India, pp. 573-576, 1999

[4] L. Teng, L.-W. Jin, Hybrid Recognition for One Stroke Style Cursive Handwriting Characters, *Proc.8th Int'l Conf. Document Analysis and Recognition*, Seoul, Korea, pp. 232-236, 2005

[5] Z.-Y. Feng, S.-J. Xu, X. Zhang, L.-W. Jin, Z.-C. Ye, W.-X. Yang, Real-time Fingertip Tracking and Detection using Kinect Depth Sensor for a New Writing-in-the Air System, *Proc. 4th Int'l Conf. Internet Multimedia Computing and Service*, Wuhan, China, pp. 70-74, 2011

[6] M. Cheriet, N. Kharma, C.-L. Liu, C.Y. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*. John Wiley & Sons, Inc., 2007

[7] Q.-F. Wang, Fei Yin, C.-L. Liu, Handwritten Chinese text recognition by integrating multiple contexts, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1469-1481, Aug, 2012

[8] X.-D. Zhou, J.-L. Yu, C.-L. Liu, T. Nagasaki, K. Marukawa, Online Handwritten Japanese Character String Recognition Incorporating Geometric Context, *Proc. 9th Int'l Conf. Document Analysis and Recognition*, Curitiba, Brazil, pp. 48-52, 2007

[9] C.-L. Liu, H. Sako, H. Fujisawa, Effects of Classifier Structures and Training Regimes on Integrated Segmentation and Recognition of Handwritten Numeral Strings, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1395-1407, Nov. 2004

[10] Q.-F. Wang, F. Yin, C.-L. Liu, Improving Handwritten Chinese Text Recognition by Confidence Transformation, *Proc. 11th Int'l Conf. Document Analysis and Recognition*, pp. 518-522, Sept. 2011

[11] M.-Y. Chen, A. Kundu, S. N. Srihari, Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition, *IEEE Trans. Image Processing*, vol. 4, no. 12, pp. 1675-1688, Dec. 1995

[12] C.-L. Liu, X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, *Proc. 10th Int'l Workshop on Frontiers in Handwriting Recognition*, La Baule, France, pp. 217-222, 2006

[13] D.-H. Wang, C.-L. Liu, X.-D. Zhou, An approach for real-time recognition of online Chinese handwritten sentences, *Pattern Recognition*, vol. 45, no. 10, pp. 3661–3675, Oct, 2012

[14] http://www.microsoft.com/en-us/kinectforwindows/

[15] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, CASIA online and offline Chinese handwriting databases, *Proc.5th Int'l Conf. Document Analysis and Recognition*, Beijing, China, pp. 37–41, 2011