

## Penmanship Learning Support System: Feature Extraction for Online Handwritten Characters

Tatsuya Yamaguchi, Noriaki Muranaka, Masataka Tokumaru

► **To cite this version:**

Tatsuya Yamaguchi, Noriaki Muranaka, Masataka Tokumaru. Penmanship Learning Support System: Feature Extraction for Online Handwritten Characters. Hyun Seung Yang; Rainer Malaka; Junichi Hoshino; Jung Hyun Han. 9th International Conference on Entertainment Computing (ICEC), Sep 2010, Seoul, South Korea. Springer, Lecture Notes in Computer Science, LNCS-6243, pp.496-498, 2010, Entertainment Computing - ICEC 2010. <10.1007/978-3-642-15399-0\_71>. <hal-01055589>

**HAL Id: hal-01055589**

**<https://hal.inria.fr/hal-01055589>**

Submitted on 13 Aug 2014

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Penmanship Learning Support System: Feature Extraction for Online Handwritten Characters

Tatsuya Yamaguchi<sup>1</sup>, Noriaki Muranaka<sup>2</sup>, Masataka Tokumaru<sup>2</sup>,

<sup>1</sup> Kansai University Graduate School, 3-3-35 Yamate-cho, Suita-shi, Osaka, 564-8680 Japan,

<sup>2</sup> Kansai University, 3-3-35 Yamate-cho, Suita-shi, Osaka, 564-8680 Japan,

E-mail: {k526329, muranaka, toku}@kansai-u.ac.jp

**Abstract.** This paper proposes a feature extraction method for online handwritten characters for a penmanship learning support system. This system has a database of model characters. It evaluates the characters a learner writes by comparing them with the model characters. However, if we prepare feature information for every character, information must be input every time a model character is added. Therefore, we propose a method of automatically extracting features from handwritten characters. In this paper, we examine whether it correctly identifies the turns in strokes as features. The resulting extraction rate is 80% and in the remaining 20% of cases, it extracted an area near a turn.

**Keywords:** Penmanship, Evaluation of characters, Features, Turns

## 1 Introduction

Classes and correspondence courses on penmanship are very popular. In the penmanship classes, the learner attends there regularly. Therefore, the time to attend there is needed, and it is not suitable for a busy person. In a correspondence course, the learner learns the shape and balance of a character according to a model character in the text and the accompanying explanation at favorite time. Afterward, the learner sends the teacher his fair copy of the character, and the teacher corrects it and sends the result back to the learner. This is the general flow of a correspondence course, which lacks real-time feedback due to the time required for sending materials. Therefore, we developed a penmanship learning support system that automatically evaluates characters. The purpose of this system is to reproduce the environment of the penmanship classroom at home. In other words, the learner can learn penmanship freely without being limited in place and time. In this research, we extract features from the model character and that produced by the learner. We then evaluate the characters by comparing the length of features and the angles between them. In this paper, we describe the feature extraction method. We want to extract the beginnings, endings, turns, and curves of the strokes. We do not describe the hardware and software here because we describe them in [1].

## 2 Feature Extraction

The character data has coordinates at 10 ms intervals. We call each of the points located at these intervals a Point. We extract the points that become feature candidates from the Points and then define the selected Points as features.

First, we describe the method of extracting the points that are feature candidates. It can be divided into three stages. In the first stage, we let the beginning and ending points  $P_B$  and  $P_E$ , respectively, of the stroke be feature candidates. In the second stage, we search for the feature parts. We describe the procedure below.

1-1. Replace  $P_B$  with a base point  $P_b$ .

1-2. Replace the second Point from  $P_b$  with a moving point  $P_m$ .

1-3. For all Points that exist between  $P_b$  and  $P_m$ , evaluate the distance  $D_P$  between it and the straight line  $P_b P_m$ . Evaluate the Point  $P_{\max}$  and the distance  $D_{\max}$  for which  $D_P$  is the maximum.

1-4. If  $D_{\max}$  is more than  $\delta_1$  (which is 2 in this paper), let  $P_{\max}$  be a feature candidate point. Replace  $P_{\max}$  with  $P_b$ . If the second Point from  $P_b$  is  $P_E$ , end. Otherwise, move to step 1-2.

1-5. Otherwise,  $P_m$  is moved to the following Point. If  $P_m$  is  $P_E$ , end. Otherwise, move to step 1-3.

In the third stage, we let the turns in a stroke be feature candidate points. We show the extraction method below.

2-1. Evaluate  $\angle FC_i P_{ij} FC_{i+1}$  of the adjoining feature candidate points  $FC_i$ ,  $FC_{i+1}$  and Point  $P_{ij}$  that exists between them. ( $1 \leq i \leq N_{FC} - 1$ ,  $1 \leq j \leq N_P$ ;  $N_{FC}$  is the number of feature candidate points, and  $N_P$  is the number of Points that exist between  $FC_i$  and  $FC_{i+1}$ .)

2-2. For all  $j$ , find  $P_{ij}$  satisfying  $\angle FC_i P_{ij} FC_{i+1} < \delta_2$ . ( $\delta_2$  is a threshold; a value of 100 is used in this paper.) If it is not found, move to step 2-5.

2-3. Let distance  $D_{P_{ij}}$  denote the minimum of the two distances between  $FC_i$  and  $P_{ij}$  and between  $FC_{i+1}$  and  $P_{ij}$ .

2-4. Let  $P_{ij}$  for which  $D_{P_{ij}}$  is the maximum be feature candidate points.

2-5. For all  $i$ , repeat steps 2-1 to 2-4.

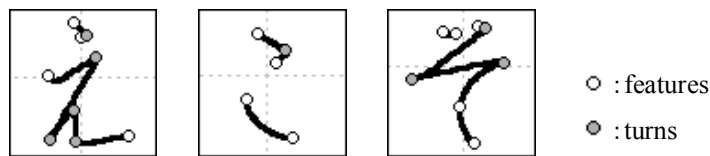
Next, we describe the method of deleting an unnecessary feature candidate point. We examine three consecutive feature candidate points  $FC_{i-1}$ ,  $FC_i$ , and  $FC_{i+1}$  and delete  $FC_i$  that matches the following deletion condition:

3-1.  $\angle FC_{i-1} FC_i FC_{i+1}$  is more than  $100^\circ$ .

3-2. The distance between  $FC_i$  and the straight line  $FC_{i-1}FC_{i+1}$  is less than seven pixels.

### 3 Experimental Results

We extracted features from the 46 hiragana characters. Fig. 1 shows an example of the resulting feature extraction from three characters. Table 1 shows the resulting extraction of turns in strokes.



**Fig. 1.** Example of feature extraction result (“え,” “こ,” and “そ” from the left).

**Table 1.** Feature extraction of turns in strokes.

Number of turns in strokes	Number of extractions	Extraction rate (%)	Average distance of the error extraction (pixel)
80	64	80	1.88

The average distance of the error extraction is the average of the difference in the distance between the turns of 16 features extracted by mistake and the actual features. It can be said that the turns will almost be extracted because they are gaps of less than two pixels.

### 4 Summary

In this paper, we proposed a feature extraction method for online handwritten characters and examined the turns in strokes that it extracted. The resulting extraction rate is 80%. In the remaining 20% of the cases, it has extracted the area near a turn in a stroke (the average error distance is 1.88 pixels).

### References

1. Sawamoto, T., Hayano, Y., Muranaka, N., Tokumaru, M.: The Penmanship (script learning) Support System, Human Interface Symposium 2008, pp. 383—388 (2008) (in Japanese)