6 ICLEI 2016-57 Andrew Chiou

Real-time Handwriting Recognition Interface for Teaching and Learning of Languages in a Multilingual Society

Andrew Chiou
School of Engineering & Technology,
Central Queensland University,
Rockhampton, Queensland, Australia
Corresponding Author: a.chiou@cqu.edu.au.

ABSTRACT

This paper details the technical design and implementation of a real-time multilingual handwriting recognition interface for teaching and learning of languages in a multilingual society. This implementation allows multiple handwritten languages using Romanised characters (e.g. English and Malay) and ideographs (e.g. Chinese and Japanese) to coexist on a single system for teaching of handwriting. The paper describes how characters written in any language can be dissected into segments of basic strokes. This is mathematically proven that each character stroke, in any language used, have a numerical property that can be uniquely encoded. Based on this encoding, a handwriting teaching software utilising this method is capable of determining if a student is writing a character accurately in the specific language being taught. The advantage of this encoding are: (1) It can be implemented independent of most graphemes and most logographies type written languages, (2) The algorithm is technically accessible and implementable using simple web based scripts such as HTML5, CSS3 and JavaScript, (3) Character accuracy is based on written dynamic stroke-accuracy, and not on static character-pattern. The paper will describe the algorithm in detail and a complete example is presented. A description of a functioning high-fidelity prototype is presented.

Keywords: handwriting recognition, handwritten characters, multilingual, graphemes, logographies.

Introduction

In Singapore and Malaysia, it is common for the English language to coexist with other languages of the local community and education system. For example, in Malaysia, in addition to the mandatory Bahasa Malaysia (National Malay Language), one other or more languages is also taught. This is often the English, Chinese and Tamil languages. With such multilingual diversity, it would benefit to have a writing input system that allows the recognition of different languages. This input system differs from conventional keyboard or handwriting input systems that allow users to create meaningful words and sentences in regular communication. The input system described in this project is specifically designed and developed for the purpose of teaching handwriting.

This paper details the design and implementation of a multilingual handwriting input system that is capable of recognising handwritten Romanised characters and ideographs. The basic features of graphemes in Romanised characters and ideographs are discussed to explain characteristics that allow them to coexist on the same system. This is followed by a detailed

explanation of the algorithm used to encode these features to allow subsequent processing and recognition.

Background

This section will discuss the basic features of Romanised characters and ideographs to explain the features that allow them to coexist on the same handwriting recognition input system. For simplicity, the words letter and character are used interchangeably.

Compare the following construction of the letter *H* (Figure 1) and the Chinese character (Figure 2) for the word *I* (me, my, mine). The average number of strokes required to write a Romanised character is three strokes, while a Chinese chracter requires an average of 10.7 strokes (Ng and Wu, 1990).

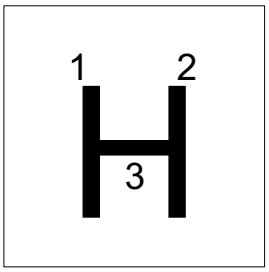

Figure 1. Romanised characters require an average of three strokes to construct.

Figure 2. Chinese ideographs require an average of 10.7 strokes to construct.

Even though the Romanised character seems easier to write than Chinese, both have similar rules that govern their construction and reproduction. That is, the construction of characters in most languages are based on the following rules:

- the number of strokes used,
- their placement in relation to other strokes,
- the direction (or vector) of individual strokes and
- the sequence of each individual stroke.

In most conventional handwriting recognition systems or apps, the processsing carried out to identify the handwritten characters is static. The image of the character is captured *after* it has been fully constructed (Plamondon and Srihari, 2000). This loses most of its unique key identifying properties even before the pattern-matching process is carried out. However, in a dynamic system, such as the type presented in this project, the identifying properties of a character are captured and preserved in real-time *while* it is being written. In this way, the characters retain their handwritten characterisitics with additional key features that will further contribute to the accuracy of the pattern-matching process. These key

features correspond with the four construction rules described above. As the construction rules is language-independent and applies equally to either Romanised characters and ideographs, different character sets from two or more languages lend itself well to coexisting on the same handwriting input system used for teaching of handwriting.

Technique

The multilingual handwriting input systems in this project is based on a modified version of Teitelman's (1964) earlier work on handwritten character recognition and systemised by Firebaugh (1988). Unique key features of a character can be captured and preserved at two levels of input: preservation of single-stroke characters and preservation of multi-stroke characters.

The encoding algorithm functions by dividing the writing area into nine zones using four grid lines labelled *A, B, C* and *D* (Figure 3). Characters that are written onto this area are encoded using a four variable schema, with each variable corresponding to the individual grid line.

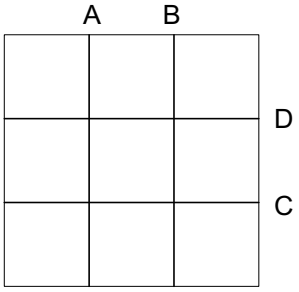

Figure 3. The writing area is divided into nine zones by four grid lines labeled *A, B, C* and *D* corresponding to the variables applied in the encoding algorithm.

The algorithm works by detecting the starting position, vector and the intersection of each stroke with the grid lines. The complete encoding algorithm is as follows:

Begin
Step 1. Initialise variable *A=0, B=0, C=0 and D=0*;
Step 2. If a stroke begins from the **left** of grid line *A*,
then *A=A+8* else *A=A+0*;
Step 3. If a stroke begins from the **left** of grid line *B*,
then *B=B+8* else *B=B+0*;
Step 4. If a stroke begins from the **below** grid line *C*,
then *C=C+8* else *C=C+0*;
Step 5. If a stroke begins from the **below** grid line *D*,
then *D=D+8* else *D=D+0*;
Step 6. Repeat step 1 to 5 until stroke is complete.

begin
For every stroke that intersects a grid line
add 1 to its corresponding variable;
e.g. if a stroke intersects grid line *B* twice, then *B=B+2*.
end
End

Preservation of Single-stroke Characters

Applying the above encoding algorithm to process the handwritten letter *L*, the resulting encoded pattern is 9-9-1-1 (Figure 4). As the letter *L* begins on the left of grid line *A* and *B*, the corresponding variables are set to *A=8* and *B=8*. And since the stroke begins above (and *not* below) grid line *C* and *D*, the corresponding variables are set to *C=0* and *D=0*. Continuing, the stroke proceeds to intersect every grid line once resulting in the increment of each corresponding variables by 1. That is, *A=A+1, B=B+1, C=C+1* and *D=D+1*. The final result of the encoded pattern of the letter *L* is thus, 9-9-1-1.

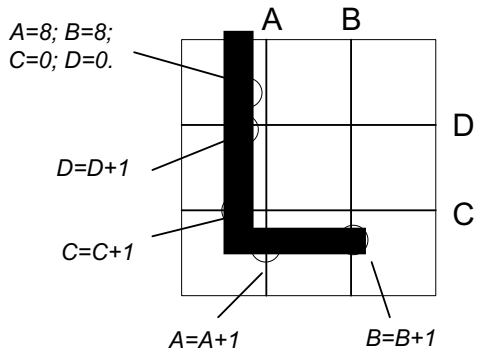

Figure 4. By applying the encoding algorithm to the letter *L*, the resulting encoded string is 9-9-1-1.

Preservation of Multi-stroke Characters

By compounding the preservation of single-stroke characters, the algorithm can be applied to multi-stroke characters. This is accomplished by appending each of the 4 digit string to form a lengthier string. For example, the letter *H* consists of three strokes (Figure 5). The first stroke is encoded into the string 8-8-1-1, the second into 0-0-1-1 and the third into 9-9-0-8. Appending the three strings into a single string we obtain, 8-8-1-1-0-0-1-1-9-9-0-8.

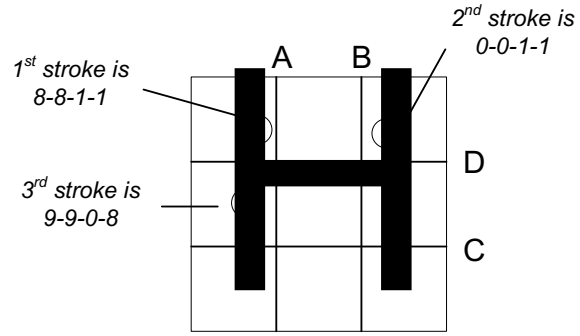

Figure 5. In multi-stroke character, the encoded pattern of each stroke is appended to the previous string. For the letter *H*, the encoded string is 8-8-1-1-0-0-1-1-9-9-0-8.

Application to Other Languages

As the encoding algorithm is language independent, the handwriting input system can be adapted to other languages with minimal modification. As a result of this, any stroke in any

ideographs based on graphemes can be uniquely encoded. In the following example (Figure 6), the Chinese character for the number *5*, is 9-9-0-0-1-8-1-1-9-9-1-8-9-9-8-8.

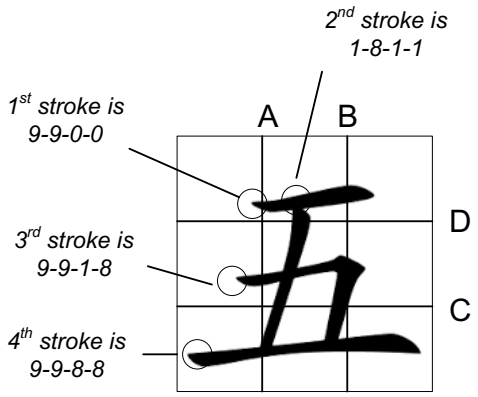

Figure 6. The encoded pattern for a multi-stroke Chinese character for the letter *five* is 9-9-0-0-1-8-1-1-9-9-1-8-9-9-8-8.

Proof

Chiou (2015) extended and used the grid system originally introduced by Teitelman (1964). It was a novel idea that produced excellent results while dealing with Romanised characters and ideographs with a few exceptions that could be handled efficiently. But when it comes to mathematical and other symbols, the number of exceptions grow significantly and it becomes impossible to identify all the exceptions and deal with them on case by case basis. The problem lies with the way the sequences are generated by taking account of the crossing of the internal grid lines only. As a result, different strokes may be created using same grid intersections and vice versa. Here we propose a different labeling scheme of the same grid that will result in unique sequences for unique strokes. We call these sequences, Lye sequences for Lye's original contribution in Chiou (2015). The following diagram (Figure 7) reflects our scheme.

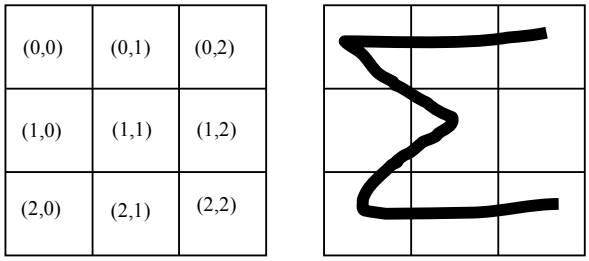

Figure 7. Grid and zone.

The above is an example of a single stroke character. Its Lye sequence as given by Definition 2 is (0, 9, 7, 2). This sequence is stored in a knowledge base as a standard notation for the Greek letter Sigma commonly used in mathematics for summation. For more precision each grid square can further be divided into a 3×3 grid and so on for the system to be trained for a specific user.

Any character or symbol drawn in this system can be uniquely coded using the following technique and then decoded relatively easily.

Definition 1. A simple horizontal stroke is a stroke starting from a grid square (i,j) and ending at (i,j±1) and a simple vertical stroke is the one starting from (i,j) and ending at (i±1,j).

Definition 2. A simple Lye sequence of a simple horizontal stroke is a tuple (i, 2i, 2j ± 1, j ± 1) and that of a simple vertical stroke is (i, 2i ± 1, 2j, 2j) starting from the grid square (i,j).

Definition 3. A Lye sequence is a tuple (r, I, J, c), where r is the row number of the starting point, I is the sum of all the row numbers appearing in the stroke, J is the sum of all the column numbers appearing in the stroke and c is the column number of the end point of the stroke.

Definition 4. Length of a Lye sequence is the number of simple strokes in the sequence.

Using the above definitions we conclude the following.

Theorem 1. A simple Lye sequence is unique.
Proof. It is easy to see from Definition 2 that all four sequences for left, right, up and down simple strokes are different.

Now we can prove that each distinct stroke is uniquely identified by Lye sequences.

Theorem 2. A Lye sequence is unique.
Proof. We prove this using induction. A simple Lye sequence has length 1 and it is unique as proved in Theorem 1. Now consider a unique Lye sequence (r, I, J, c) of length k with the end point at (i,c). We can extend this sequence to a sequence of length k+1 by adding one simple stroke in three of four possible directions. For a horizontal move we get (r, I + i, J + c ± 1, c ± 1) and for a vertical extension we get (r, I + i ± 1, J + c, c). As we see all these four extensions are unique. Hence any Lye sequence is unique.

Thus, we conclude that any character or symbol composed of multiple strokes will also generate a composite Lye sequence.

Results

Based on the encoding algorithm, the multilingual handwriting input system interface prototype was subsequently developed and tested. The working high-fidelity prototype was developed to be accessible from the web. It is written in HTML5, CSS3 and JavaScript. A pointing device, such as a pen was used to write the desired character. As the application is meant as a teaching tool, a mouse and fingertip is not recommended. A database of the alphabets of the English and Malay written characters were encoded. As both these two languages shared a common letter set, this was a convenient matter of re-labeling the teaching instructions in either English or the Malay language. A second database was created for 50 characters of the Chinese language (characters with an average of 7.5 strokes). The input system works with both sets of database simultaneously. That is, a student can write in any language presented by the application. As the system is capable of differentiating the different characters, the teaching instruction presents the language of choice to the students.

In massive real time data gathering of user feedback, conventional data collection methods such as questionnaires and surveys are impractical and inefficient for the purpose of this project. Therefore, a recent and novel method of data gathering has been designed through the use of crowd sourcing (McDuff et al, 2011). During Open Day and Expo organised by the author at several of its campuses, the prototype was made freely available to the public for direct hands-on trial. The public composed mostly of adults from diverse background and school children. The application commences by demonstrating how a character is written either in the English, Malay or Chinese language. The user respond by attempting to duplicate the character shown by writing using the touch pen provided.

Language	Accuracy
English	95%
Malay	96%
Chinese	99%

From the results an interesting observation emerges. In languages that rely on complex number of strokes to construct, the accuracy of the real-time handwriting input system yields a higher score. This can be attributed to the length of the encoded string. That is, the uniqueness and accuracy of the characters feature that are preserved increases as the number of strokes increases.

Summary

In a multicultural and multiethnic diverse society, where different languages are used interchangeably in day to day communication, the medium to teach the different languages should reflect the same level of adaptability. Hence, the real-time handwriting recognition input system capable of recognising handwritten characters in Romanised characters and ideographs have been introduced. This will allow educators to be able to implement a technically simple, but accessible method of designing and developing a similar system into their instructional software that are web based or as an app. This paper has detailed the algorithm on how unique features of handwritten characters can be encoded and preserved as key identifiers. This is language independent, allowing for different languages to coexist on the same system. To demonstrate the feasibility of the algorithm, a working prototype has been developed.

References

Chiou, A. (2015), “Real time handwriting recognition techniques for mathematical notation in interactive teaching & learning applications”, *Journal of Advanced Research in Applied Mechanics*, vol. 15, no. 1, pp. 20-26,

Firebaugh, M. W. (1988). *Artificial Intelligence: A Knowledge-Based Approach*, PWS-Kent Publishing Company.

McDuff, D., El Kaliouby, R., Picard, R. (2011). Crowdsourced data collection of facial responses. *Proceedings of the 13th international conference on mul-timodal interfaces - ICMI’11, 2011.*

Ng, Y. H. and Wu, W. H., (1990), “Learning to Write Chinese From First Principles”, *Computers & Education*, vol 15 (1-3), pp.119-126.

Plamondon, R. and Srihari, S. N. (2000). "On-line and off-line handwriting recognition: a comprehensive survey*", IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 63–84*.

Teitelman, W. (1964), *Real Time Recognition of Hand-drawn Characters*, Spartan Books.