Penerbit Akademia Baru

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

Real Time Handwriting Recognition Techniques for Mathematical Notation in Interactive Teaching & Learning Applications

A. Chiou

School of Engineering & Technology, Central Queensland University, Rockhampton 4701
Queensland, Australia.
a.chiou@cqu.edu.au

Abstract – *This paper presents a selection of primary techniques used in the encoding of handwritten mathematical notation into a meta-string that can later be converted to a standard document markup language. This method is used to archive or deliver documents with handwritten mathematical notation via the Internet utilising platform independent devices. The techniques used include, encoding sequence that are unique for each symbol; and fuzzy membership functions to determine position, spacing and size of symbols.*

Keywords: Intelligent computing, handwriting recognition, pattern mapping, internet.

1.0 INTRODUCTION

This paper presents a selection of primary techniques employed in recognising handwritten mathematical notation and encoding it as a meta-string that can be further converted to a standard document markup language. This is an update of the author's previous work and a report on the enhancement that have been introduced in Human Computer Interaction course delivered at Central Queensland University [7].

This project is an extension to the working prototype completed by Chiou and Lye [2]. The real time handwriting recognition expert system (HRES) was first developed as part of a module in a computer-assisted language learning system environment to instruct foreign languages [3]. Throughout its research and development, it was discovered that the inference engine used by HRES to recognise Romanised characters in languages such as Bahasa Malaysia and English; and other ideograph-based languages such as Chinese, Japanese and Arabic, can also be used to process mathematical symbols.

At Central Queensland University (CQU), the mode of instruction has been a gradual shift from traditional classroom based delivery to Internet delivery. However, a number of challenges arose from using non-printed material. One such challenge was the transmission of documents containing handwritten mathematical symbols via the Internet. Arguably, one can use a document markup language such as LaTex or HTML; or an equation editor such as MathType to produce such documents. But such editing utilities have a steep learning curve and are not always accessible to either lectures or students. Other methods used for delivery such as, facsimile, scanned documents and normal mail were also proven to be slow and cumbersome.

Penerbit Akademia Baru

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

This project employs the use of an input-interface that allows the user to write mathematical notation while an intelligent module processes the input concurrently. Unlike conventional handwriting recognition systems, HRES processes handwritten input in real time, allowing users to instantly observe what has been written and corrections to be made. Once the input is complete, the characteristics of the handwritten notation are encoded into a meta-string. The meta-string will consist entirely of standard ASCII characters that can be conveniently transmitted or archived using platform-independent devices. This meta-string can then be subsequently used to reproduce the original mathematical notation intended.

This new development provides an opportunity to further extend an earlier work on an intelligent system for online tutoring [4].

2.0 CHARACTER RECOGNITION

In [2] Chiou and Lye extended and used the grid system originally introduced by Teitelman [5]. It was a novel idea that produced excellent results while dealing with Romanised characters and ideographs with a few exceptions that could easily be handled. 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 labelling 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 [2]. The following diagram (Fig.1) reflects our scheme.

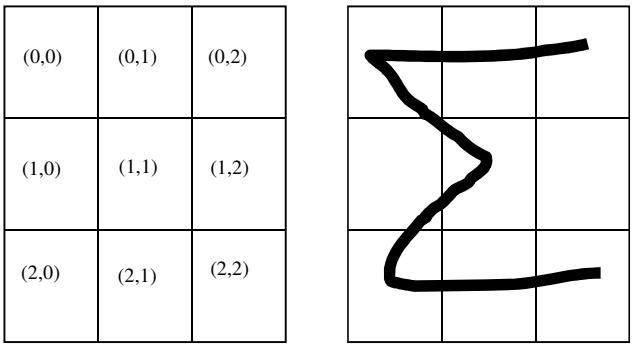

Figure 1: Grid System

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 either in LaTeX or HTML format relatively easily.

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

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.

3. META-STRING

Mathematical notations are fundamentally composed of strings, S. That is,

$$S = \{Z, Z_m, D\} \tag{1}$$

where,

$$Z = \{x \mid x \in \textit{alphanumeric symbols}, \; x \in \textit{operator symbols}, \; x \in \textit{normal-size}\} \tag{2}$$

$$Z_m = \{y \mid y \in \textit{alphanumeric symbols}, \; y \in \textit{operator symbols}, \; y \in \textit{minor-size}\} \tag{3}$$

$$D = \{\$, \&, \hat{}, \#\} \tag{4}$$

S is made up of normal-size in (2) and minor-size (proportionally diminished version of normal-size symbols) in (3). Minor-size symbols are used to denote subscript and superscript.

Penerbit Akademia Baru

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

S also include delimiter symbols, *D* in (4), to denote grouping order of sub-strings ($ and &), superscript (^) and subscript (#).

For example, the expression

$$f(x) = \frac{1 + x^2 - 2x^2}{\left(1 + x^2\right)^2}$$

is converted to

S = $f(x) $= $&1 $+ $x^2 $- $2x^2& $/ $&($1 $+ $x^2 $)^2& (5)

where the delimiter, $, denotes the preceding of each sub-string in *S,* as in (5). The delimiter, &, marks the preceding and proceeding of a sub-string to denote the numerator and denominator parts of a fraction. Here, *S* suffice being the meta-string - which can be further translated into other document markup languages. The process is simply reversed to reproduce the original expression into a human-readable format.

The following sections explain a selection of techniques used in the processing and formation of *S*.

3.1 Encoding Primary Features

To convert a handwritten mathematical sequence, three primary features of each input character, $C_{n,}$ must first be determined before it can be appended to *S*. These features are, symbol size, spacing and position.

The input-interface is a writing surface divided into grids to form a matrix. This matrix is subdivided into sectors resembling the grid system in Figure 1. As each symbol or character is written onto the input-surface, the inference engine in HRES will process and attempt to recognise the features of these input. Each symbol that has been successfully encoded will be stored as part of *S*.

The basic conversion algorithm (Figure 2.) shows how an expression can be progressively appended to *S* after the current input character, $C_{n,}$ has been determined by Theorem 2. As (1) shows that *S* is composed of *Z* and Z_m, once characteristic of C_n has been determined to be of relative normal-size or minor-size symbols compared to the previous character C_{n-1}, the appropriate delimiter, # and ^ can then be appended to denote subscript and superscript respectively.

Penerbit Akademia Baru

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

```
Case Cn = size_normal:

   If  Cn-1 = space then
         S = $ + Cn
   Else
         S = S + Cn
   Endif

Case Cn = size_minor and
          superscript:

   If Cn-1 = size_normal then
      S = ^ + Cn
   Else
      S = S + Cn
   Endif

Case Cn = size_minor and
          subscript:

   If Cn-1 = size_normal then
      S= # + Cn
   Else
      S = S + Cn
   Endif
```

Figure 2: Algorithm to form meta-string *S*.

As inherent in all handwriting, characteristics of handwritten symbols can vary from individual to individual. Hence, a fixed unit of measurement used to determine the characteristics of handwritten symbols are inflexible. In this case, if-then rule-base decision [1] using fuzzy sets [6] are employed. The fuzzy membership functions to evaluate the current symbol C_n are,

$$\mu_{\sup}(x_n;h_{n-1})=\begin{cases}0 & \text{for} \quad x_n<\dfrac{h_{n-1}}{2}\\[2ex] \dfrac{x_n-\dfrac{h_{n-1}}{2}}{\dfrac{h_{n-1}}{2}} & \text{for} \quad \dfrac{h_{n-1}}{2}\le x_n\le h_{n-1}\\[2ex] 1 & \text{for} \quad x_n>h_{n-1}\end{cases} \tag{6}$$

and

Penerbit Akademia Baru

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

$$\mu_{\text{sub}}(x_n;h_{n-1})=\begin{cases}0 & \text{for} \quad x_n>\dfrac{h_{n-1}}{2}\\[2ex] \dfrac{\dfrac{h_{n-1}}{2}-x_n}{\dfrac{h_{n-1}}{2}} & \text{for} \quad 0\le x_n\le\dfrac{h_{n-1}}{2}\\[2ex] 1 & \text{for} \quad x_n<0\end{cases} \tag{7}$$

where for both (6) and (7),

x_n = distance of C_n from base level of C_{n-1}

and

h_{n-1} = height of previous character, C_{n-1}.

Hence, depending on the degree of fulfilment (DOF) each membership function would yield, C_n is appended to S as a superscript or superscript. Likewise, other fuzzy membership function can be used to evaluate the DOF for spacing and the relative size of symbols using if-then rules.

3.2 Encoding Fractions

The fraction symbol presents one of the more complex challenges in handwritten mathematical notation. This symbol, unlike its other operator counterparts (ie. +, -, (,)), is not fixed and varies in length in each notation. The method used is to keep track of the numerator and denominator parts as delineated by the fraction symbol.

Figure 3 shows how to handle the non-linearity of notation employing fractions. As the numerator (S_1) and denominator (S_2) is separated by a horizontal line, sub-string delimiters (&) are used. Once the entire fraction element composing of S_1 and S_2 is obtained, it is appended to S. In this way, a recursive function can be used to encode nested fraction notation.

```
Case Cn = fraction_symbol:

   loop
      get all numerator Cn
      S1 = S1 + Cn
   end loop

   S  = $& + S1 + &

   loop
      get all denominator Cn
      S2 = S2 + Cn
   end loop

   S =  $/ + $& +  S2
```

Figure 3: Algorithm to encode fraction notation.

Journal of Advanced Research in Applied Mechanics
ISSN (online): 2289-7895 | Vol. 15, No. 1. Pages 20-26, 2015

One of the methods used by HRES to keep track of the length of the horizontal line is to register the starting and ending x-coordinates of the line. Then, by comparing the coordinates of all symbols written above the line, all the symbols consisting of the numerator can be obtained. Similarly, the denominator is obtained by comparing all the coordinates of symbols written below the fraction line.

4. OUTCOME

We have developed a real time mapping system for hand written characters and mathematical symbols for an interactive intelligent system for a computer assisted learning package. The proposed theory and techniques can be extended to image processing systems for both the real time and static images. A preliminary prototype is working and a more sophisticated version is under construction.

REFERENCES

[1] R.C. Berkan, S. L. Trubatch, Fuzzy Systems Design Principles: Building Fuzzy IF-THEN Rule Bases, NY: IEEE Press, 1997, ch. 3, pp. 83-129.

[2] A. Chiou, N.C. Lye, A Handwriting Recognition Interface for a Multilingual Society: A Working Prototype for Simultaneous Real-time Recognition of Handwritten Romanised Character and Ideographs, Proceedings of the First Asia Pacific Conference on Computer Human Interaction (APCHI '96), Singapore, June 1996, pp. 192-202.

[3] A. Chiou, D. Clayton, P. Farrands, N.C. Lye, A Creative Mandarin Tutor: Some Project Experiences, Proceedings of the Twelfth Annual Conference of the Australian Society for Computers in learning in Tertiary Education (ASCILITE '95), Melbourne, Australia, 1995, pp.57-62.

[4] M.A. Salam, An Intelligent NetTutor, Proceedings of the 5th Pacific Rim Symposium on Natural Language Processing (NLPRS'99), Beijing, China, 1999, pp. 455-458.

[5] W. Teitelman, Real Time Recognition of Hand-drawn Characters, Spartan Books, 1964, ch. 14, pp. 456-462.

[6] L.A. Zadeh, Fuzzy Sets, Information and Control 8 (1965) 338-353.

[7] COIS 12036 Human Computer Interaction (HCI) Term 1 2015, 2015, Moodle Course Website, Central Queensland University.