Matching Algorithm for Hangul Recognition Based on PDA

Hyeong-Gyun Kim, Gwang-Mi Choi, *Member, KIMICS*

Abstract—Electronic Ink is a stored data in the form of the handwritten text or the script without converting it into ASCII by handwritten recognition on the pen-based computers and Personal Digital Assistants(PDA) for supporting natural and convenient data input. One of the most important issue is to search the electronic ink in order to use it. We proposed and implemented a script matching algorithm for the electronic ink. Proposed matching algorithm separated the input stroke into a set of primitive stroke using the curvature of the stroke curve. After determining the type of separated strokes, it produced a stroke feature vector. And then it calculated the distance between the stroke feature vector of input strokes and one of strokes in the database using the dynamic programming technique.

Index Terms—PDA, electronic ink, Hangul matching.

I. INTRODUCTION

There have been various research reports on the data processing and recognition for the electronic ink of PDA since the early 1990s. As the effective way of searching electronic ink data from the large data base, several approximate ink matching algorithms for the Hangul recognition of electronic ink data have been proposed [3][4], and methods utilizing the blind Markov Network and R-tree rate have been proposed[5][6]. However, these methods were developed with the main input type of English script. As compared writing English and Hangul scripts with pen, the English scripts can be decomposed into the continuous circular arcs, while the Hangul has the clear cornering position and direction of strokes. The proposed algorithm for matching the English scripts had been implemented[4] and applied for the Hangul scripts, showing very low matching rates. Consequently there is no meaning in directly applying the matching algorithm for the English scripts to the Hangul scripts, so that it is required to apply a Hangul matching algorithm for the electronic ink data mainly with the Hangul scripts, considering the geometric features of Hangul.

This study proposes the Hangul matching system for storage and recognition in the form of electronic ink using the script data input with pen in the Hangul PDA.

Manuscript received July 22, 2004.

Hyeong-Gyun Kim is with field of Computer and Internet, Dongkang College, Gwangju, Korea.(e-mail : multikim87@hanmail.net)

Gwang-Mi Choi is with field of Computer and Internet, Dongkang College, Gwangju, Korea.(e-mail : iplab@hanmail.net)

II. RELATED WORK

Given a pictogram, whether it is an item to be inserted into the database, or a query that is used for retrieval from the database, we need to transform it to a more robust form before passing it to the database system for further processing. (The original representation of the pictogram, which comes in terms of a series of z, y coordinates is too vulnerable to slight changes. This makes it innapropiate for matching.) Traditional databases use an alphanumeric representation of the data. This representation is ideally unique, precise and stable. Ink data lacks these qualities, making its matching a difficult problem. The data is often corrupted with noise. Even ideal ink does not provide an adequate basis for sequence identification, because we want to be able to identify a pictogram given slightly different variations in its shape. You cannot find two people that write the same word in the same way. Even for the same person, it is very difficult to generate exactly the same pictogram twice (it will almost always be the case that the stroke information varies each time the person writes the same word). The first issue that needs to be settled is the selection of the granularity that is to be represented in the database.

In Figure 1, we show a pictogram and its segmentation into pen strokes and handwritten symbols. We can choose to represent pictograms by any of the three granules presented in the figure, i.e., as one entity containing the entire pictogram (Figure la), as a sequence of pen strokes (Figure lb), or as a sequence of alphabet symbols (Figure lc). Of course, in order to select, for instance, the symbols as granules, we have to have a segmentation algorithm that properly separates the symbols. For instance, for strokes, a simple segmentation algorithm picks local minimum (or maximum) points and uses them to segment the curve

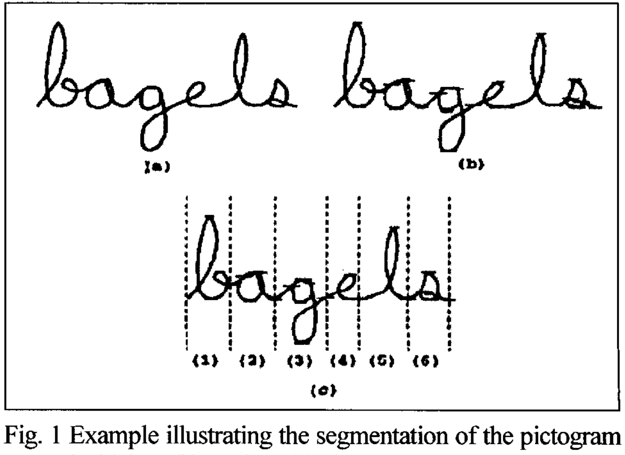

Fig. 1 Example illustrating the segmentation of the pictogram in (a) into (b) strokes, (c) alphabet symbols.

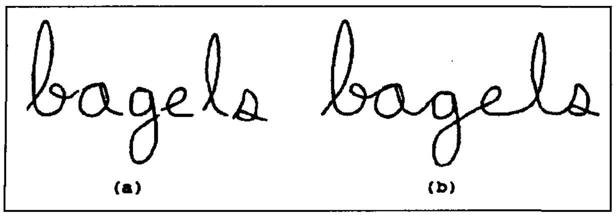

Fig. 2 An example of (a) a handprinted word, cursive handwritten word.

Segmentation could be a difficult task for some types of pictograms, such as cursive handwritten words (see Figure 2b), or a simpler task as in handprinted words (see Figure 2a). Some languages, like Japanese, lend themselves easily to symbol segmentation. In Japanese, Kanji symbols are already separated by blank spaces, The choice of granularity has an impact on the type of indices we build.

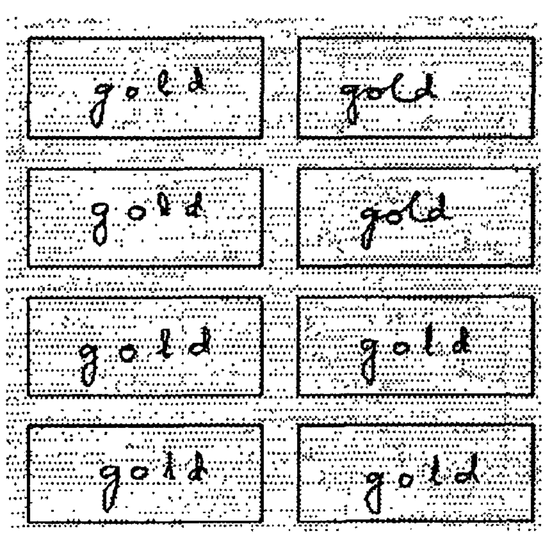

Fig. 3 A list of pictograms, all representing concept “gold”.

The second issue is that, for matching purposes, it is better to talk about pictogram (symbols, strokes) classes instead of individual pictograms. A pictogram class is the set of pictograms that have the same semantics, according to the user. Figure 3 shows a (non-exhaustive) list of pictograms that represent the concept “gold.” Of course, it would be impractical to store a pictogram class by storing the list of pictograms that belong to it. Alternatively, we have to choose a representative of the class and a distance metric.

Inputs are matched against the representative (after the necessary preprocessing) and the distance metric is used to rank the matches. The representative is not a pictogram, but rather a model that captures the essential qualities of the class. To build the representative model one needs to use features of the pictograms that belong to the class. Features can be of two types. 1 Local Features: One way of representing ink is to pick some sample points from the pictogram (or symbol, or stroke) and compute some local features. By local, we mean that the computed features depend on a sample point and possibly the one (or two) surrounding points from each side. It is important to mention that, depending on the application, some of these features may be more relevant than others, and hence not all of them need be computed at a given point in the sequence. Common features are: direction, velocity, change in direction, change in velocity, accumulative angle (with respect to the initial point), accumulative length and angle of bounding box diagonal (also with respect to the initial point), and accumulative sequence length.

The input format of our pictogram is an array s of P time-stamped sample points:

$$s_p = (x_p, y_p, t_p),\ 0 \leq p < P \qquad (1)$$

After computing the local features, the pictogram can be represented by a sequence of feature vectors (v_1, t_1), (v_2, t_2),...The dimensionality of the vi corresponds to the number of local features at each point, Global Features: Global features are characteristics of the entire pictogram. Among them are: the bounding box coordinates, the total angle traversed by the pictogram (measured by the angle from the beginning point to the endpoint), and length of the bounding box diagonal. After computing global features, the pictogram can be represented by a vector of global feature values. Having collected a set of features for the pictograms (symbol;or strokes), we can proceed to model the- lass. We show here how to build two representative models. The choice of model also has an impact on the index technique that we use.

A. Hidden Markov Models

HMMs are already used in the field of speech and handwritten recognition as a powerful tool for speech and handwritten document matching.[2] Each pictogram in the database can be modeled by an HMM. The HMM is constructed so that it accepts the specific pictogram with high probability (relative to the other pictograms in the database). In order to recognize a given input pictogram, we execute each HMM in the database and select the one that generates the input sequence with the highest probability, Since each HMM in the underlying sequence database has to be tested, this results in a linear process where the speed of execution is the primary difficulty. An HMM is a doubly stochastic process, where there is a probability distribution that governs the transitions between states and an output probability distribution that identifies the distribution of output symbols for each state.

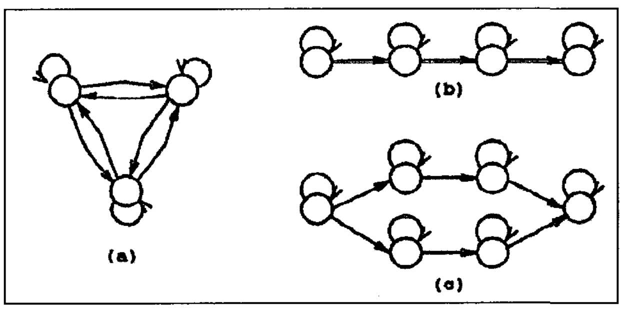

Fig. 4 Several examples of HMMs: (a) the ergodic model with three states, (b) a left-to-right model with four states, and (c) a parallel path left-to-right model with six states.

We use one type of HMM structures, termed left-to-right HMM[4] (e.g., see Figure 4b). The left-to-right type of HMMs is useful for modeling temporal signals as in sound and cursive handwritten text because the underlying state sequence associated with the model has the property that, as time increases, the state index increases (or stays the same) — that is, the system states proceed from left to right. A left-to-right HMM can be constructed to model a handwritten word or an alphabet symbol. The HMM is constructed so that it accepts the word (or the symbol) with high probability (relative to the other words in the database). 1 The probability y given by the HMM is the distance metric used for ranking purposes. Given an HMM that models a word (or symbol), we can run an input symbol against it and obtain as an output a matching probability, Given a set of stored words (or symbols), one can match an input word (or symbol) by running each one of the corresponding HMMs against the input and choosing those with the best matching probability. In fact, we can keep the size of the answer set as a parameter and choose the k best matches.

B. Indexing Ink

As we shall see shortly, sequential searching does not scale well. The process becomes unacceptably slow as the number of items stored grows. Therefore, indexing techniques that help pruning the choices are extremely helpful. However, indexing techniques for handwritten pictograms must exhibit two characteristics that makes them different from traditional indexing techniques:

- The structures must incorporate the underlying model that is chosen to represent ink. The model must play an active part of the index.
- Due to the high variability of the ink data, the indices must provide approximate matching.

This paper describes one such technique, termed by us the handwritten trie, which exhibit both of the characteristics stated above. As we mentioned before, the choice of granularity and representative model dictates the type of index that we build. For instance, we can choose to model entire pictograms with HMMs and build indices that use the HMM characteristics to guide the search (we call such an index the HMM-tree [2]). Alternative y, we can choose to deal with alphabet symbols for granularity and represent the symbol classes by using HMMs. The handwritten trie, which will be described in the next section, uses the second approach.

III. PROPOSED HANGUL MATCHING ALGORITHM

In this study the Hangul matching algorithm was used to implement a system so that the prime script of Hangul is used in the form of the electronic ink data in the PDA. First divide the ink data into basic stroke units by the Hangul matching algorithm and then apply the dynamic programming technique. While considering hardware limitation of PDA, it is designed to store the Hangul numeric data on the CF memory to enhance the recognition rate as well as to secure the effective speed. When converting Hangul data value by various electronic ink data on the PDA, the applied system matches the converted one with Hangul with the existing Hangul data.

A. Preprocessing

Shaking hand may cause mistakes on curves and strokes when writing. Mistakes are compensated through smoothing, filtering, removing mistakes and normalizing sizes.

$$X_i = \frac{(X_{i-3}+3X_{i-2}+6X_{i-1}+7X_i+6X_{i+1}+3X_{i+2}+X_{i+3})}{27} \tag{2}$$

The current value is displaced by an average with the weight around 7 points, and the stroke curves are smoothed as shown in Eq. (2). The irregular spacing of points due to writing speeds are finalized uniform through DDA(Dot Density Algorithm) filtering. In this study the directional vector has been featured to skip the other preprocessing.

B. Feature extraction process

The process extracts the feature and virtual vectors [5][6] from the coordinate data of the enhanced stroke as the information on letter element recognition. The input strokes would show various forms of scribbles along with irregular lengths and angles in view of writing styles. So the feature points are utilized as the basic recognition unit to reduce unessential volume of information among the coordinate data column of input strokes[2][3]. However, several feature points may exist in specific parts of input strokes according to the writing speeds and angle variations, so that the process of removing rings, decorative lines, and neighboring feature points removes unnecessary feature points congregated within the given distance of marginal values. The feature and virtual vectors are information to match with the Hangul database, so that the feature vectors are procedural and directional information for the input strokes and the virtual vectors are information for position relation between strokes in the letter element or between letter elements[2][3]. But unless many potential letter elements occur when recognizing letter elements, the virtual vector uses only the Tail-Head Vector for the faster processing. In addition, the position relation and inclusive relation of position information as position information between strokes are extracted for letter element separation and recognition. The inclusive relation is the indication of adjacent level among the minimum circumscribed squares of strokes neighboring the minimum circumscribed square including the current strokes, composing inclusion, overlap, and separation.

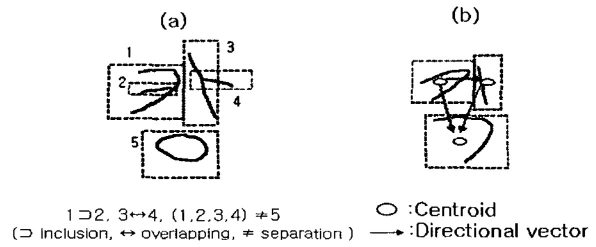

Fig. 5 Extraction of position relation among strokes

Fig. 5(a) shows a case of extracting inclusive relations for the input letter '강'. Dotted squares are the minimum circumscribed squares including strokes, and a number is the stroke number inside the dotted square, which means the dotted square 1 indicates the whole size for the stroke 'ㄱ'. It is indicated that the dotted square 2 is included in the dotted square 1, the dotted squares 1,3, & 5 are separated each other, and the dotted squares 3 & 4 are overlapped. The position relation is a centroid of the current stroke and the direction information among the centroids of all input strokes, and is utilized as position relation information among the letter element, inside and letter space. Fig.5(b) is an extraction case of input letter '거', and shows the direction information on centroids among each letter elements comprising of letter. The position relation among strokes provides the position to decompose into the basic strokes from calculating the curvature for the points comprising strokes when considering input strokes as one curve. When indicating curvatures over the marginal value for the arbitrary points comprising strokes, the stroke is separated at that point.

C. Determination of basic stroke units

Separation of letter element is prerequisite process for letter element recognition, and separation and recognition are in parallel processing in this study. The letter element recognition recognizes the letter element by matching stroke information extracted from the letter element separation and stroke information stored in the Hangul database and converts information of each recognized letter element into the assigned value of the electronic ink data to generate a letter.

```
Step 1. Sequential letter element separation
Step 2. Recognition of initial sound
    if (Recognize success) step 3
    else Apply back-tracking, Retry initial sound recognition
Step 3. Recognition of vowels
    if (Recognize success)
        if ( No initial sound exists)
            Complete letter element separation,
              character recognition
        else step 4
    else Back-tracking application, step 3
    if (Retrial success) step 4
    else Back-tracking application, step 2
Step 4. Recognition of final consonant
    if (Recognize success)
        Complete letter element separation, character recognition
    else Back-tracking application, step 3
        if(Retrial of final consonant recognition success)
            Complete letter element separation,
            if (Retrial of final consonant recognition success)
                Complete letter element separation
                Character recognition
                else Back-tracking application, step 2
        else Back-tracking application, step 2
```

Fig. 6 Back-tracking letter element separation algorithm

There are sequential letter element separation using position relation among strokes and back-tracking letter element separation performing recognition by changing the number of strokes comprising each letter element through matching Hangul database. The sequential letter element separation recognizes a case of separate inclusive relation among strokes as the stroke from the other source, and case of overlapped or inclusive as the strokes from the same letter element. But in case of separate inclusive relation among strokes inside letter element, the sequential letter element separation shows insufficient letter separation capability, so that the back-tracking letter element separation is performed in case of false recognition or non-recognition. The existing back-tracking method shows the demerits of reprocessing all procedures. But the back-tracking letter element separation, based on stroke information extracted, separates the total strokes of input letter into the optimal number of strokes to recognize according to the recognition by letter element. Fig.6 and Fig.7 shows an algorithm for the back-tracking letter element separation and a parallel processing for letter element separation and recognition, respectively. This study shows fast processing speed by performing the letter element separation differently according to configuration information of input letter and stroke types.

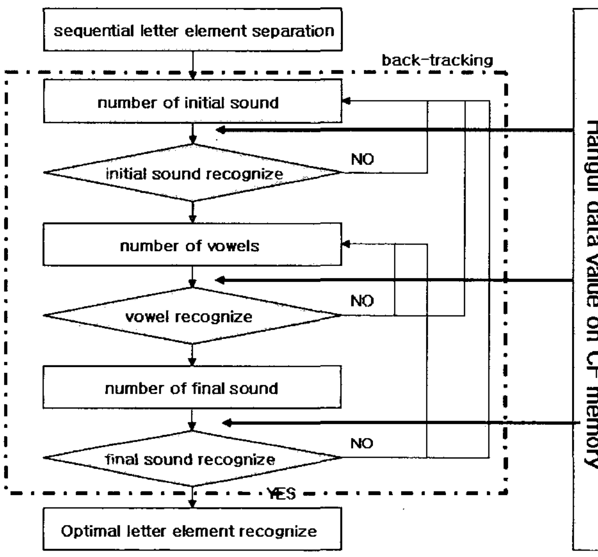

Fig. 7 Processing flow for Letter element separation and Recognition

D. Similarity calculation

The process separates the input data into the basic stroke unit by stroke curvatures. Seven basic strokes have been assigned to ensure fast matching speed in the PDA in consideration of the geometric characteristics of Hangul as shown in Fig.7. The next step decides types of basic strokes separated by the curvatures to create stroke feature vectors. The stroke feature vector has basic stroke unit information separated from the input data. Once the stroke feature vector created, the dynamic programming calculates distances among the stroke feature vectors for the stroke feature vector of query and the ink data in the database, and compares the data having the shortest distance with the data value on the CF memory to return matching result.

E. Report on matching result

The letter element recognition comprises the synchronization process of values on the Hangul database on the CF memory, containing structural configuration information of Hangul and diverse stroke information of various users for each letter element. The structural configuration information of Hangul includes packing information among letter element using in recognizing the vowels, collection information, position relation among strokes, letter element classification information by configuration, etc. In addition, letter element models for recognition by letter element has the general stroke information. However, the letter containing excessive curvatures and any scribbles between letter elements is difficult to extract the accurate information, then utilizing only straight line, smooth curve information and stroke configuration in various directions. Fig.8 shows the hierarchical recognition process for the Hangul data ink.

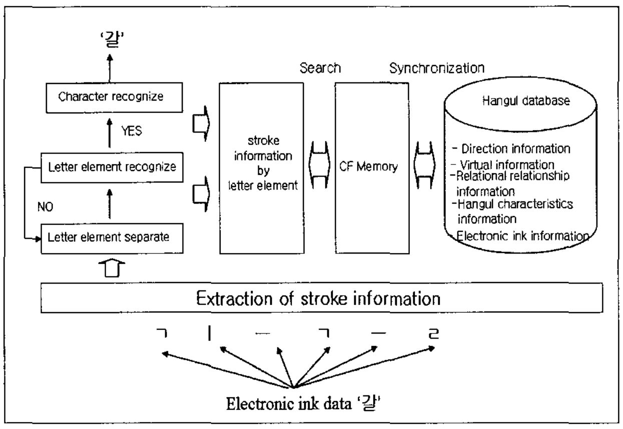

Fig. 8 Hierarchical recognition process

IV. IMPLEMENTATION AND CONSIDERATION

The randomly extracted electronic ink data collection was used for the test, and input type was the Hangul scripts. Visual C# and MS-SQL were applied to construct Hangul data on the CF memory, and the mobile program was prepared in Embedded Visual C++ and performed with the test PDA(Compaq ipaq 3850). Matching rates were measured with the three element, and the model is as follows.

$$R = m(n,t,r) \tag{3}$$

The test results show merits of the matching algorithm as follows. First, the matching algorithm is practical. Generally the number of the electronic ink data using in the PDA and mobile computer is around 300. The suggested algorithm shows the matching rate of nearly 98% even with 300 database. Second, the fast searching speed shows an average 0.8 second in matching speed with 300 database, which is a very fast speed when considering the environment of the PDA having hardware limitations as compared with the desktop computer. Third, one merit of writer-dependent matching algorithm provides the security capability for anybody having different strokes. In case of the writer-dependent matching, the different strokes for the same letter recognize the different form of pattern. Therefore when anyone of different strokes wants to use, the probability of providing desired data becomes low, providing the security capability for the private information.

V. CONCLUSION

This study suggested and implemented the Hangul matching system for storing and recognizing the electronic ink form with the Pen-input script data in the PDA. The suggested Hangul matching system calculates the stroke curvature along with the preprocessing procedure and works through the process separating into the basic stroke units. Then it creates the stroke feature vectors by the decision of types of basic stroke unit for calculating the distance value with the dynamic programming. When calculating the distance value, the edit operation includes deletion, insertion, and exchange operation as well as packing and separation operation in consideration of the characteristics of Hangul. The resulting values are used to compare the Hangul numeric value on the CF memory and recognize Hangul, applying it to the system.

The matching algorithm suggested in this study shows the matching rate near 98% for the Hangul scripts even with 300 database. In addition, it shows the average matching speed of 0.8 second with 300 database, which is very fast speed in consideration of the PDA environment with hardware limitation as compared to the desktop computer. Next, the merit of the writer-dependent matching algorithm would provide the security capability for anyone having different strokes.

Next topics would be as follows; first, how to make fictitious problem in data efficient for effective construction of Hangul database on the CF memory and to apply to the system, and second, how to carry in the user electronic ink data on the PDA, to understand characteristics of each users and to synchronize to the database value on the CF memory. The final topic is to find a method to enhance the matching rate in the input stage without classifying letters when reading in scripts.

REFERENCES

[1] W. Aref, D. Barbarii, and P. Vallabhaneni. The Handwritten Trie: Indexing Electronic Ink. Technical report, M. I.T.L, October 1994.

[2] Walid Aref and Daniel Barbar& The Hidden Markov Model Tree Index: A Practical Approach to Fast Recognition of Handwritten Documents in Large Databases. Technical Report MITL-TR-84-93, MITL, January 1994.

[3] Walid G. Aref, Padmavathi Vallabhaneni, and Daniel Barbar& Towards a realization of handwritten databases: Training and recognition. Technical Report MITL-TR-98-94, Matsushita Information Technology Laboratory, Princeton, NJ, March 1994.

[4] R. Bakis. Continuous speech word recognition via centisecond acoustic states. In PTOC. ASA Meeting, Washington, DC, April 1976.

[5] Daniel Barbar& Method to index electronic handwritten documents. Technical Report MITL-TR-77-93, Matsushita

Information Technology Laboratory, Princeton, NJ, November 1993.
[6] R. Carr and D. Shafer. The Power of PenPoint. Addison-Wesley, 1991.
[7] K. Landau, S. Major, and C. Wiederhold. The Role of PDA in the Office. Notes of the Seminar at PC-EXPO, June 1994.

Hyeong-Gyun Kim
He received the M.S and Ph.D. degrees in the Dept. of Computer Engineering from Chosun University.
His research interests include Multimedia, Image processing, Mobile communication, Char acter recognition.

Gwang-Mi Choi
She received the M.S. and the Ph.D. degrees in the Dept. of Computer Science and Statistic, Chosun University. Her research interests Multi media, Neural networks, Multimedia contents, Artificial intelligence, Information security, Networks.