

Word-matching method based on the projection of the voting matrix

T. Akagi, T. Hamamura, H. Mizutani, B. Irie
TOSHIBA Corp.,
70 Yanagi-cho, Saiwai-ku, Kawasaki-shi, Kanagawa 212-8501, Japan
E-mail: *takuma.akagi@toshiba.co.jp*

A new word-matching method is proposed that is able to achieve matching speedily and correctly even in the presence of noise at the beginning, end or within the word. This method can use the discriminant function in order to choose the most fitting word from the database. These functions take into consideration factors such as the character recognition rate and word segmentation rate. In addition, the method can determine the difference in noise between the word in the database (reference word) and the word extracted from the character line and recognized (character-recognition result).

1 Introduction

Word matching in this paper is defined as the ability to find the word in the database that is the most similar to the character-recognition result which has been extracted and recognized from character lines. This technique is used in various products such as mail-sorting systems.

Word matching presents a problem in that the input data include various noises made during “word segmentation,” “character segmentation” and “character recognition.”

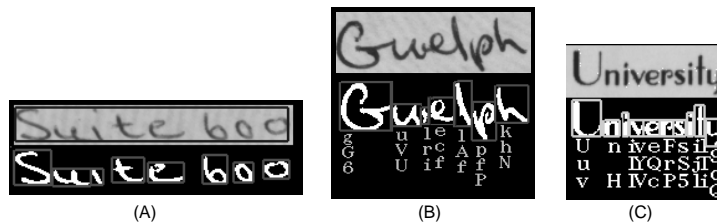


Figure 1: Input word data including noise at the end

The authors present these problems by referring to examples of incorrect address reading. Shown in Fig.1 (A) is an example of the incorrect extraction of a word due to “suite” and “600” have been extracted as a single word. To recognize addresses that include numbers, the system must be robust against noise. Moreover, it must be able to distinguish the matching letters from those that do not match.

Shown in Fig.1 (B) are the word data GUELPH and the character candidates for it (rectangles). The lower three lines of characters are the results of character recognition that are highest in rank (ranked in descending order from uppermost to lowermost). Because the character segmentation is unsuccessful, there is an incorrect character candidate between “u” and “e”. To match this result to the reference word, it is necessary for the system to be able to consider the noise within the character-recognition result.

Shown in Fig.1 (C) is an example of incorrect recognition. “r” and “t” are only second-ranked candidates and “y” is not a candidate. In order to be successful, the system must have flexibility with respect to incorrect character recognition, and must also be able to find several character-recognition candidates.

The conventional string matching methods are unable to consider the noise within the word data¹²³⁴⁵. In some matching methods, incorrect character recognition is assumed⁶⁷, but these methods use only one character-recognition candidate. If these methods were to use several candidates, matching would be immensely time-consuming. On the other hand, voting is used in some methods. However, since it is difficult to consider the sequence of letters, matching rates are low; for example, it is difficult to distinguish between “TOWNWEST” and “WESTTOWN.” Accordingly, it is desirable for the matching process to have the merits of speed, consideration of the sequence of letters, robustness against noise and consideration of the preprocess methods.

2 Matching considering the sequence of letters

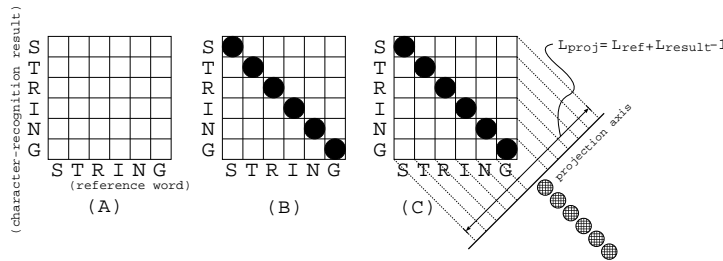


Figure 2: The voting matrix for matching

In this section, an outline of the proposed method is presented. First, the matrix is derived. The vertical line indicating the character-recognition result and the lateral line indicating the reference word form a complete matrix. Shown in Fig.2 (A) is an example of a matrix where the character-recognition result is “STRING” and the reference word is also “STRING.” Next, the voting

on the matrix is done if the character-recognition result and the reference word have a letter or letters in common. The position for voting is the intersection point of the letter which the character-recognition result and the reference word have in common (Fig.2 (B)). Next, the projection axis is made in the direction from upper right to lower left. The contents of the projection axis are the projection made diagonal to the voting matrix (Fig.2 (C)). In this case, the maximum value of the projection is 6 and its position is at the center of the projection axis (maximum vote position). L_{ref} is the length of the reference word, L_{result} is the length of the character-recognition result and L_{proj} is the length of the projection axis.

The normalized value of the maximum value of the projection indicates the similarity between the matched words. An example of this normalization is presented below.

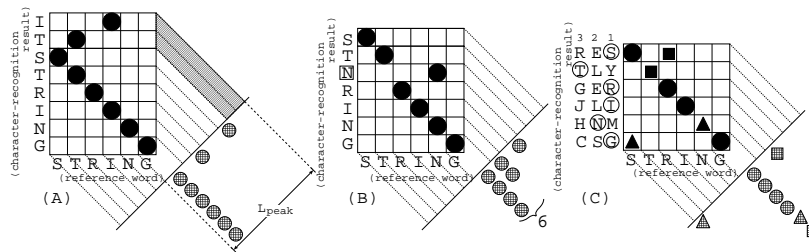


Figure 3: The variation of the voting

Now, let us consider the case in which the noise “IT” attached to the beginning of the character-recognition result. Fig.3 (A) shows the result of the projection of the voting to the matrix when the character-recognition result is “(IT)STRING.” The character-recognition result and the reference word differ in length, thus the matrix is a rectangle. The maximum value of the projection is 6. The noise letters “I” and “T” are included in the voting, but they do not have a great influence on the projection. It is found that the maximum vote position has shifted from the center of the projection axis, because the extra boxes corresponding to noise letters “I” and “T” are attached to the end of the projection axis. Thus, by counting the number of boxes from the maximum vote position to the end of the projection axis, it is possible to ascertain the position and number of the noise letters attached to the ends of the character-recognition result. The number of noise letters attached to the beginning of the character-recognition result can be expressed as $(L_{peak} - L_{ref})$. On the other hand, the number of noise letters attached to the end of the character-recognition result can be expressed as $(L_{proj} - L_{peak} - (L_{ref} - 1))$.

For the case of the noise letter “N” inside the word “STRING,” Fig.3 (B)

of the i th class on the database. x should be classified into the class which makes the *a posteriori* probability $p(w_i|x)$ maximum. Then the *a posteriori* probability can be changed according to the Bayes theorem as follows:

$$p(w_i|x) = p(x|w_i)p(w_i)/p(x). \quad (1)$$

The *a priori* probability $p(w_i)$ can be obtained beforehand. The term $p(x)$ is the factor common to the classes. Thus, $p(x|w_i)$ is proportional to $p(w_i|x)$. Then $p(x|w_i)$ can be expressed as

$$p(x|w_i) = C \times P_1^{\alpha_1} \times P_2^{\alpha_2} \times P_3^{\alpha_3} \times P'^{(L_{ref}-\alpha_1-\alpha_2-\alpha_3)} \times P_{nosf}^f \times P_{nosm}^m, \quad (2)$$

where C is the constant for normalization, P_k is the character recognition rate at the k th candidate, α_k is the number of the projection of the voting of k th candidates and P' is the possibility that the correct letter does not show up as one of the (“third” in this case) candidates. P_k and P' can be obtained when the character recognition method is selected. P_{nosf} (or P_{nosm}) is the possibility that the noise letters are attached to the end of (or inside) the character-recognition result and f (or m) is the number of noise letters.

As explained above, α_k , f and m are already known when the voting is finished, and $p(w_i|x)$ can be immediately obtained.

5 Experimental Results

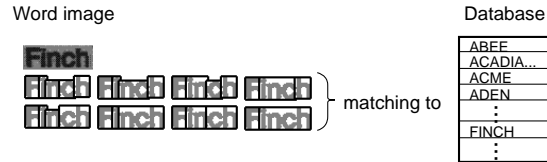


Figure 5: The matching data

To evaluate our method, we carried out an experiment with character-recognition results extracted from character lines. A total of 537 word images are used as the input data. Each of the word images has several candidates according to the segmentation of the character candidates (Fig.5). On the other hand, a total of 1210 of the reference words registered in the database are used. The average length of reference words is 9 letters.

Table 1 shows a comparison of the matching rate of our method for the top word with that of the common voting methods for the top 5 and top 20 words. Our method can detect or reject one candidate using discriminant function (2). The matching rate of our method is higher than those of other methods.

	time(sec)	match(%)	err(%)	reject(%)
our method (Top)	22.91	84.1	0.7	15.2
voting (In top 5)	16.07	64.1	-	-
voting (In top 20)	16.07	78.7	-	-

Table 1: The result

6 Conclusion

Matching methods can be divided into methods based on voting and methods based on a searching of the path. Voting methods can quickly match a word but present a problem in that it is difficult to consider the sequence of letters in the words. Methods based on a searching of the path can consider the sequence of letters in the words, but it is time-consuming to consider various orders of character recognition and noise attached to the word. In this paper, we proposed a new method that can match the words using the voting, and thus retains the given sequence of letters. Moreover, this method can use the discriminant function based on factors such as the character recognition rate in order to choose the most fitting word from the database.

References

1. D. E. Knuth, J. Morris and V. Pratt, "Fast pattern matching in strings," *SIAM Journal on Computing*, vol. 6, no. 2, pp. 323-350, 1977
2. R. Boyer and S. Moore, "A fast string searching algorithm," *Communications of the ACM*, vol. 20, no. 10, pp. 762-772, 1977
3. Z. Liu, X. Du and N. Ishii, "An improved adaptive string searching algorithm," *Softw. -Pract. Exp. (UK)*, vol. 28, no. 2, pp. 191-198, 1998
4. J. Kärkkäinen, "Suffix cactus: a cross between suffix tree and suffix array", *Combinational Pattern Matching 6th Annual Symposium, CPM95 Proceedings (Germany)*, pp. 191-204, 1995
5. K. Marukawa, H. Fujisawa and Y. Shima, "Evaluation of information retrieval methods with output of character recognition based on characteristic of recognition error", *Trans. Inst. Electron. Inf. Commun. Eng. (Japan)*, vol. J79D-11, no. 5, 1996
6. R. H. Wagner and M. Fischer, "The string-to-string correction problem," *J. Assoc. Comput.*, vol. 21, no. 1, 1974
7. S. V. Rice, J. Kanai and T. A. Nartker, "An algorithm for matching OCR-generated text strings," *Int. J. Pattern Recognition Artif. Intell. (Singapore)*, vol. 8, no. 5, 1994