Proceedings of the 2013 Federated Conference on
Computer Science and Information Systems pp. 671–674

Fuzzy-Based Multi-Stroke Character Recognizer

Alex Tormási and László T. Kóczy
Széchenyi István University
1 Egyetem tér, Győr, H-9026, Hungary
Email: {tormasi, koczy}@sze.hu

***Abstract*—In this paper an extension for multi-stroke character recognition of FUzzy BAsed handwritten character Recognition (FUBAR) algorithm will be presented. First the basic concept of a single-stroke version will be overviewed; in the second part of the paper the new version of the algorithm with multi-stroke symbol support will be introduced, which deploy the same algorithm overviewed in the first part and use flat and hierarchical rule bases.**

I. Introduction

LALOMIA defined the user acceptance threshold in 97% [1], however most multi-stroke character recognition methods known from the literature that are applicable for 26 symbols are well below, on the other hand with a stricted set of symbols (16 gestures) the $N recognizer reached 96.7% [2]. Despite the high accuracy these methods are not always usable for on-line (real-time) handwriting recognition as a result of their high computational complexity and processing time. It is very important to find a recognition engine, which is able to process the input strokes in a short period even on devices with limited resources such as tablets.

In this paper we present a new attempt to recognize multi-stroke letters (26 symbols) with rather good recognition rate (however definitely below LaLomia's 97% threshold). As the starting point the FUBAR algorithm that was very successful for single-strokes will be used, with modifications towards multi-stroke symbols (up to 3 strokes).

After the introduction in this paper the basic steps of the single-stroke FUBAR (Fuzzy Based Recognition) algorithm [3] will be overviewed. In Section 3 the results of the new method with the capability of recognizing multi-stroke symbols are presented. In Section 4 the average recognition rates are analyzed, for the case of the same multi-stroke recognition method with a hierarchical rule-base. In Section 5 the results of the new algorithms and other known recognition algorithms are compared.

II. The Single-Stroke Fuzzy Based Recognition Method

A. Algorithm Properties and Features

During the design of the algorithm the most important goal was to create a recognition engine which is able to process the input strokes with at least the same accuracy of other already published recognition methods, while taking less computational time. Most of the methods published in literature are using geometrical transformations, like rotation and trapezoid correction are complex and resource consuming; to reduce the complexity of the recognition method the use of such transformations was ruled out. The designed recognition method is online, which means that it uses digital ink information to represent the strokes. The alphabet used there is based on a slightly modified version of the Palm's Graffiti single-stroke symbol set [4].

B. Input Processing

The method collects the positions of the digital pen used during the writing process; the strokes are stored in a time ordered list of two-dimensional coordinates. To provide a better input for the next phases of the processing, the input stroke should be re-sampled to provide a low-level anti-aliasing (noise reduction) and provide almost equal distance between the sampled points.

Details of the input processing phase are overviewed in [3, 5].

C. Parameter Extraction

The algorithm uses two kinds of parameters to recognize symbols; the first is the width/height ratio of the stroke while the second type is formed by the average numbers of stroke points in the rows and columns of the fuzzy grid drawn around the stroke. The fuzzy grid approach was introduced in details in [3], in order to handle italics and thus replace the stroke rotation phase used by other methods. The rows and columns of the grids are represented by fuzzy sets [6], which allow a point to belong to two different rows or columns at the same time.

D. Inference

The parameters extracted from a collection of 60 single-stroke character samples were used to determine the rule base for the fuzzy system.

Each symbol in the used alphabet is described by a single rule. The antecedents of the rule are the previously collected parameters and the consequent part represents the degree of matching.

For inference a discrete Takagi-Sugeno method [7] with standard t-norms was used. The algorithm returns with the symbol assigned to the best matching rule.

This paper was supported by the National Scientific Research Fund Grant OTKA K75711 and OTKA K105529, a Széchenyi István University Grant and EU grants TÁMOP 4.2.1 B, TÁMOP 4.2.2/B-10/1-2010-0010.

978-1-4673-4471-5/$25.00

The detailed description of the single-stroke algorithm and inference can be found in [3, 5].

III. Multi-Stroke Symbol Support

In this part, the results of a new Fuzzy-Based Recognition Engine (FUBAR) family member are presented. This new method is able to process multi-stroke symbols. The method handles each symbol as one non-continuous stroke with "empty" spaces between the sampled points.

The samples were collected from 10 male and 10 female Hungarian participants in the age group 18 to 40. Each subject has provided 20 samples for each 26 multi-stroke symbol. The etalon writing style for the symbols was determined by pre-collecting samples for the most widely used symbol types in the local area for the 26 letters of the English alphabet.

Symbols with obvious errors were identified and removed from the collection and the number of samples per symbols was limited to 180 for a better comparison with the single-stroke system, in which there is the same number of (different) samples was used during the tests in [8, 9, 10].

A similar method was used for data collection as overviewed in [8, 9, 10] with a modification to transform multiple strokes into a single one. This approach gives the capability to the algorithm to process multi-stroke input.

After this step the joint stroke is re-sampled to ensure almost equal distance between the neighbor points.

Each normal trapezoidal fuzzy set describing the stroke parameters in the antecedent part of the rules was constructed according to the statistical process of the parameters extracted from the first 60 samples for each symbol. Each letter is represented by one rule, where the input parameters are collected from a fuzzy grid drawn around the multi-stroke symbol. The analyzed stroke parameters are the width/height ratio and the average number of points in the rows and columns of the fuzzy grid drawn around the stroke (described in previous section); the details of the feature extraction method are the same as in [3, 5].

The same method was used for data collection as overviewed in [3, 5, 8]. The output parameters of the rules are representing the degree of matching between the parameters of the processed input stroke and the information stored in the rules for the described letter.

For the inference a discrete Takagi-Sugeno method was used with standard t-norm (Zadeh t-norm), which returns the symbol assigned to the rule with the highest matching value for the input stroke.

The average recognition rates for the symbols and for the complete alphabet were calculated for the method by the results for 120 samples from the sample set using different fuzzy grid sizes. These are the same conditions as overviewed in [8, 9, 10].

Similar single-stroke method was used for data collection as described in [3, 5], which may give a better environment to compare the results.

The best result for the multi-stroke alphabet was achieved by the algorithm using a 3x4 fuzzy grid; the letter-wise average recognition rates are listed and compared with the results of the similar single-stroke method with various modifications in Table I.

As shown in Table I, the accuracy of the system for the multi-stroke alphabet is 93.4% which is 6.03% and 5,42% less than the results of the single-stroke method (using 6x6 and 6x4 fuzzy grid).

The results have been analyzed in depth including the search for the reason of the false results. Each recognition process that has returned with false result could be traced back to the fuzzy sets describing the rule antecedents. It means there were no false-positive results caused by the overlapping sets of different parameters; and the accuracy might be increased by redefining the rule base.

IV. Hierarchical Rule-Base for Multi-Stroke Alphabet

There are many papers dealing with the use of hierarchical rule bases in fuzzy systems in different areas [11, 12, 13]. A previous work presents the results of the single-stroke method using hierarchical rule base. The details of building the hierarchical rule structure by rule input parameters for single-stroke alphabet were presented in [10].

In the modified system with multi-stroke capability the numbers of strokes were used to determine the meta-level of rules, selecting the group of rules consisting of the given number of strokes. During the tests the same data were used as in the previous section.

This method processes only the selected rules, this way the number of evaluated rules is reduced. The average recognition rates for multi-stroke and single-stroke systems are presented in Table I.

The accuracy of the system for multi-stroke alphabet is the same with flat and hierarchical rule bases. It can be explained by the results presented in the previous section in which it has been stated that there were no false-positive results and all the mistakes were caused by the limited rule base and not by an overlap of the different symbols.

V. Conclusions and Future Work

It was shown that, after the modification, the system was able to recognize multi-stroke alphabets with 93.4% average recognition rate. The results indicate that the accuracy might be further increased by a redefinition of the initial rule base.
Finally in this work a similar method with multi-stroke alphabet support using hierarchical rule base was presented. The topology of the hierarchy was built based on the number of the used strokes. The modified system reached the same accuracy as the original one with flat rule base, but the computational cost of the recognition process was considerably reduced by the limited number of rules to evaluate.

The results of the previously introduced altered FUBAR methods are shown and compared to other commercial and academic methods in Fig. 1.

As you may see in Fig. 1 the recognition rates of FUBAR for single-stroke alphabet are higher than other systems.

TABLE I.
AVERAGE RECOGNITION RATES OF THE ALGORITHM FOR SINGLE-STROKE AND MULTI-STROKE ALPHABETS

Symbol	Average Recognition Rates of FUBAR (%)				
	Single-Stroke FUBAR with 6x6 fuzzy grid	*Single-Stroke FUBAR with 6x4 fuzzy grid*	*Multi-Stroke FUBAR with 3x4 fuzzy grid*	*6x4 Single-Stroke FUBAR with hierarchical rule base*	*3x4 Multi-Stroke FUBAR with hierarchical rule base*
A	100	100	96.1111	100	96.1111
B	95.5555	92.7374	89.4444	92.7374	89.4444
C	99.4444	97.7654	76.6667	97.7654	76.6667
D	98.3333	98.8827	96.6667	98.8827	96.6667
E	99.4444	97.2067	92.2222	97.2067	92.2222
F	100	100	96.1111	100	96.1111
G	100	100	97.2222	100	97.2222
H	100	100	95.5556	100	95.5556
I	97.7777	100	98.3333	100	98.3333
J	100	100	97.7778	100	97.7778
K	100	99.4413	96.6667	99.4413	96.6667
L	100	100	98.3333	100	98.3333
M	100	100	96.1111	100	96.1111
N	100	96.0894	96.6667	96.0894	96.6667
O	97.7777	93.8548	92.7778	93.8548	92.7778
P	100	100	92.7778	100	92.7778
Q	98.8888	100	97.7778	100	97.7778
R	98.3333	100	87.7778	100	87.7778
S	100	100	97.7778	100	97.7778
T	100	100	96.1111	100	96.1111
U	100	97.2067	93.3333	97.2067	93.3333
V	100	98.3240	88.3333	98.3240	88.3333
W	100	100	92.2222	100	92.2222
X	99.4444	98.3240	89.4444	98.3240	89.4444
Y	100	99.4413	91.1111	99.4413	91.1111
Z	100	100	85	100	85
Average	**99.4231**	98.8182	*93.3974*	98.8182	93.3974

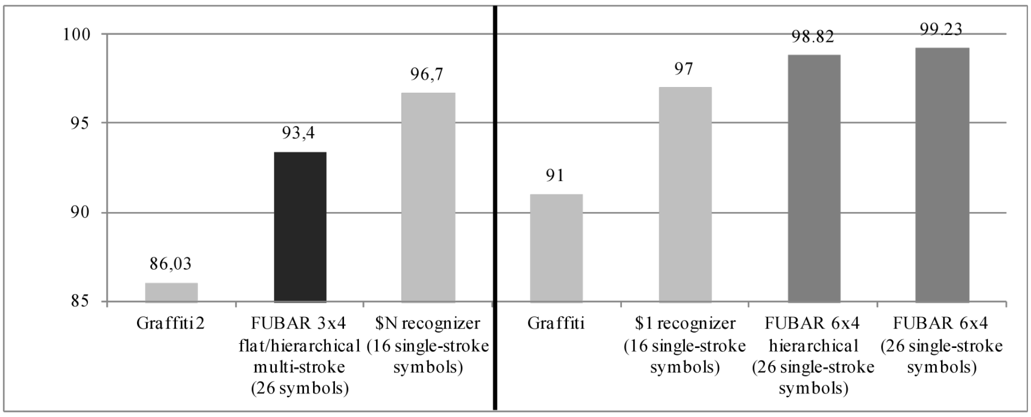

Fig. 1 Average recognition rates of various recognition engines (multi-stroke engines on the left, single-stroke engines on the right)

The $1 recognition method reached 97% average accuracy for only 16 different symbols [14], while the single-stroke version of the designed system reached over 99% recognition rate for 26 different symbols. Another advantage of the new algorithm is the computational cost of the recognition, which is linear in each phase.

Fleetwood et al. showed that users could reach only 91% average recognition rate with the original Graffiti recognition method [15], which is less than the accuracy of the presented system. The Graffiti alphabet contains 26 different English letters and other control symbols.

The average recognition rate of the modified Palm Graffiti with limited multi-stroke support (known as Graffiti 2) was studied by Költringer and Grechenig in [16]. The method reached only 86.03% accuracy. Both the single-stroke and multi-stroke versions of FUBAR performed well over the results of Graffiti 2.

The $N recognizer [2] (the multi-stroke version of the $1 method mentioned above) achieved 96.7% average recognition rate for 16 different symbols.

It is important to highlight the fact, that the computational complexity of the proposed recognition engine is linear, while most of the commercial and academic systems have a quadratic or higher computational complexity. This means that, other systems need much more time to compute the results even if the alphabet is extended by one symbol. The computational cost of FUBAR method increases however only linearly.

Currently we are working on a new method to build the initial rule base for the system, which may increase the recognition rate of the algorithm.

Another extension for the method is under development, in which the output rules are presented by discrete type-2 fuzzy sets. The preliminary results of the test are showing that the recognition rate can be increased by the mentioned modification without a significant increase of the computational cost.

In the future we intend to investigate an extension of the present algorithm, where the possibility of applying two or more rules representing a single character will be considered.

REFERENCES

[1] M.J. LaLomia, “User acceptance of handwritten recognition accuracy,” Companion Proc. CHI ’94, New York, 1994, p. 107.

[2] L. Anthony and J.O. Wobbrock, “A Lightweight Multistroke Recognizer for User Interface Prototypes,” Proc. GI 2010, Ottawa, 2010, pp. 245–252.

[3] A. Tormási, and J. Botzheim, “Single-stroke character recognition with fuzzy method,” New Concepts and Applications in Soft Computing SCI, vol. 417, V.E. Balas et al. (eds.), 2012, pp. 27–46.

[4] A. Butter and D. Pogue, Piloting Palm: The inside story of Palm, Handspring, and the birth of the billion-dollar handheld industry, John Wiley & Sons, Inc., New York, 2002

[5] A. Tormási, and L.T. Kóczy, “Comparing the efficiency of a fuzzy single-stroke character recognizer with various parameter values,” Proc. IPMU 2012, Part I. CCIS, vol. 297, S. Greco et al. (eds.), 2012, pp. 260–269.

[6] L.A. Zadeh, “Fuzzy sets,” Inf. Control, 8:338-353, 1965.

[7] T. Takagi, and M. Sugeno, “Fuzzy identification of systems and its applications to modeling and control,” IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, pp. 116-132, 1985.

[8] A. Tormasi, and L.T. Kóczy, “Efficiency and accuracy analysis of a fuzzy single-stroke character recognizer with various rectangle fuzzy grids,” Proc. CSCS ’12, Szeged, 2012, pp. 54–55.

[9] A. Tormási and L.T. Kóczy, “Improving the Accuracy of a Fuzzy-Based Single-Stroke Character Recognizer by Antecedent Weighting,” Proc. 2nd World Conference on Soft Computing, Baku, 2012, pp. 172-178.

[10] A. Tormási and L.T. Kóczy, “Improving the Efficiency of a Fuzzy-Based Single-Stroke Character Recognizer with Hierarchical Rule-Base,” Proc. 13th IEEE International Symposium on Computational Intelligence and Informatics, Óbuda, 2012, pp. 421-426.

[11] M. Sugeno, F.M. Griffin, and A. Bastian, “Fuzzy hierarchical control of an unmanned helicopter,” Proc. IFSA ’93, Seoul, 1993, pp. 1262-1265.

[12] M. Sugeno, and K.G. Park, “An approach to linguistic instruction based learning,” Intern. Journal of Uncertainity, Fuzziness and Knowledge-Based Systems, 1(1):19–56, 1993.

[13] L.T. Kóczy, and K. Hirota, “Approximate inference in hierarchical structured rule bases,” Proc. IFSA ’93, Seoul, 1993, pp. 1262–1265

[14] J.O. Wobbrock, A.D. Wilson and Y. Li,, “Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes,” Proc. UIST '07. ACM Press, New York, 2007, pp. 159–168.

[15] M.D. Fleetwood et al., “An evaluation of text-entry in Palm OS – Graffiti and the virtual keyboard,” Proc. HFES ‘02, Santa Monica, CA, 2002, pp. 617–621.

[16] T. Költringer and T. Grechenig, “Comparing the Immediate Usability of Graffiti 2 and Virtual Keyboard,” Proc. CHI EA ’04, New York, 2004, pp. 1175–1178.