Fuzzy handwriting description language: 

FOHDEL

Ashutosh Malaviya¹, Liliane Peters*

GMD - German National Research Center for Information Technology, Schloß Birlinghoven, 53754 St. Augustin, Germany

Received 4 June 1998; accepted 14 January 1999

Abstract

In this paper we present a new fuzzy language – FOHDEL – for the syntactic description of handwritten symbols. The presented language incorporates the fuzzy logic techniques to describe the syntactic relations of the semantic features extracted from a symbol pattern. A FOHDEL rule-base represents the compact feature information extracted from a small number of character prototypes and covers various handwriting styles. After presenting the theoretical basis of the fuzzy formalism, the structure of the proposed language is illustrated with some examples. An automatic FOHDEL rule generation strategy is briefly discussed. © 1999 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Fuzzy logic; FOHDEL language; Rule-base; Handwriting recognition; Automatic rule generation

1. Introduction

Structural pattern recognition is based on the idea to describe a complex pattern in terms of a hierarchical composition of simpler sub-patterns. In syntactic pattern recognition a basic set of primitives forms the terminal set of the grammar. The pattern class is the set of strings generated by the pattern grammar [1]. But the concept of formal grammar is too rigid to be used for the representation of real-life patterns such as handwritten documents. This rigidity can be changed if a certain fuzziness is introduced which describes the vagueness of such patterns [2]. Accordingly, a fuzzy language can handle imprecise patterns when the indeterminacy is due to inherent vagueness [3]. The conventional approaches to knowledge representation usually lack the means to represent the imprecise concepts. Consequently, the first-order logic-based approaches or probability theory do not provide an appropriate conceptual framework to deal with the imprecise or vague information [4]. Humans express such vague information in terms of features and natural language [5,6]. Fuzzy sets offer a theoretical basis to cope with the vagueness of patterns, which we exploited in the proposed method. First the motivation for the development of a fuzzy language is given. This is followed by the syntax and structure of the proposed language. Then the rule building syntax is explained. The generative power of the language is illustrated by the automatic FOHDEL rule generation strategy. Other handwriting recognition approaches may be found in Refs. [7–11].

2. Development of the new language

2.1. Background and motivations

In recent years various fuzzy software and hardware system development tools [12] e.g. FuzzyClips, FUNNY- Lab [13] have been introduced.² The goal of

² http://www-isis.ecs.soton.ac.uk/research/nfinfo/fuzzy.html.
these tools is to provide a convenient and ergonomic way for configuring membership functions, defining rules, input and output functions, etc. However, these tools are not suitable for highly structured data applications such as pattern recognition. In pattern recognition the symbolic description in a compact form is more useful for analysis as well as for recognition problems [14]. Moreover the large number of input and output variables make the rule-base incomprehensible and contains redundant information also.

Parizeau et al. [15] have proposed allograph-based method to recognize cursive handwritten words with fuzzy logic. The drawback of this method is that, there is no direct way of generating the handwriting feature allographs automatically. The idea is to represent patterns in form of a compact fuzzy language. This language should be capable of reducing the pattern information into a generative syntactic structure. Primarily this helps in achieving a good amount of data compression and secondly it provides the capability of automatic rule-base generation. The purpose of introducing fuzzy logic in such a language is for providing the relaxation in description of patterns. The relaxation can be achieved by fuzzy linguistic features and combination of selected features with fuzzy operators or connectives. These connectives can be used for additional information compression [16]. The combination of fuzzy features with fuzzy operators with a predefined syntax can be called as fuzzy rules. The optimization of prototype fuzzy rules can also save the cost of recognition and avoid the redundant rules.

The summary of the above considerations is that with a new fuzzy language we have two fundamental motivations; (1) automatic generation of a pattern description in a rule-base and (2) the representation of patterns in a linguistic comprehensive form. Through the generative power of such a language its extension to other pattern recognition applications can be accomplished by changing the feature space or enlarging the rule-base [17].

2.2. Components of the proposed language

A language is a collection of sentences of finite length which is constructed from a finite alphabet or in case of syntax description it is limited to a finite vocabulary. A grammar can be regarded as a device that enumerates the sentences of a language. Formal methods of syntax and semantics are often employed to describe symbols [4,18,19]. Methods for describing characters in a linguistic form have been presented by various researchers over the last 30 years [15,19]. The existing linguistic techniques in pattern recognition are based on the structure of underlying relationships between features in a two-dimensional pattern [1]. If such a structure is identified then a complex pattern can be described in terms of basic primitives and sub-patterns. But the precision required by formal languages in pattern recognition contradicts with the imprecision or ambiguity of real-life patterns. To overcome this mismatch between the imprecise nature of the input and the precise description of the syntax it was proposed by Fu [1] to introduce an uncertainty factor or fuzziness into the structure of formal language. This led to the development of stochastic and fuzzy languages [20,21]. Following the same reasoning as Fu we have developed a new fuzzy language named “FOHDEL-Fuzzy On-line Handwriting DEscription Language” [16].

The primary components of FOHDEL language are linguistic pattern or sub-pattern features e.g. shape, size and location [22]. The linguistic definition of these pattern features can be extended with linguistic modifiers e.g. fuzzy linguistic hedges and limiters. For the combination of the modified features with other features some logical connectives will be needed e.g. fuzzy aggregation connectives [23].

The linguistic description of these components can be accomplished in a form of fuzzy rule base in a relatively simple syntax [23]. This rule-base is then helpful to estimate the belongingness of the unknown handwriting patterns in form of fuzzy membership function. The fuzzy membership of possible classification results can be computed with fuzzy measures, operators and different composition rules [24]. This means that the quality of outputs is influenced by the individual weights of these rules to the final classification answer value [25].

The new language incorporates the fuzzy logic techniques to describe the syntactic relations of the semantic features extracted from a symbol pattern. What is new about this language? Primarily it incorporates the uncertainty factor at all levels from feature description over rule generation upto the classification level. It is designed with the help of fuzzy grammar and supports the linguistic description of patterns including handwritten symbols [26]. The rule-base is compact due to symbolic prototypes and operators and can be modified or extended according to the changed situation with respect to patterns. The application area of the proposed language is wide and not limited only to the description and analysis of on-line handwriting [17].

3. Syntax and structure of FOHDEL

As explained in the last section the main idea behind FOHDEL is to have a linguistic tool with which a minimal number of character prototypes can be generated to match with a large number of samples acquired from different writers. Its roots come from the existing picture

---

3 The name FOHDEL was given because of its first application to on-line handwriting recognition.
description language (PDL) and the plex languages [14,18]. Fuzzy grammars are employed to describe the syntax of languages and these grammars can be used to represent the structural relations of patterns. Due to the additional generative power of fuzzy grammars over conventional ones, we chose an attributed fuzzy grammar for the proposed FOHDEL language. For a detailed description of fuzzy grammars properties and languages see also [1,4,27,28]. Following are the basic definitions to support the formal definition of this language.

Definition 1. A fuzzy grammar $G_F$ is a quintuple

$$G_F = (V_N, V_T, P, S_0, \{ \mu \}),$$

where $V_N$ and $V_T$ are finite disjoint sets of non-terminal and terminal vocabulary, correspondingly, such that $V = V_N \cup V_T$ is the total vocabulary of the grammar. $P$ is a finite set of production rules of the type $\{ \alpha \to \beta \}$ where $\alpha \in V_N$ and $\beta$ is member of the set $V^*$ of all strings (including the null string $\epsilon$). $S_0 \in V_N$ is the starting symbol. $\{ \mu \}$ is a mapping of $P$ into $[0,1]$ such that $\{ m(p) \}$ denotes the possibility of a defined (FOHDEL) rule $p \in P$.

Definition 2. A fuzzy language $L(G_F)$ is constructed with a fuzzy grammar as

$$L(G_F) = \{ [x, \{ \mu(x) \}] | x \in V^*_F, S_0 \Rightarrow x \}.$$  

The syntax of FOHDEL includes basic primitives (complex features), linguistic attributes and operators. The non terminal vocabulary consists of descriptions of a pattern i.e. Character. Character contains the Elements which are connected with the terminal set of Operators and together it has an attributive linguistic Membership. The Elements can be modified with fuzzy Modifiers. If there are more than one elements then they are necessarily combined with fuzzy Operators. The combination of elements results in the membership of the whole character (See Example 1). The selection of terminals and non-terminals depends upon the application and takes place in problem configuration phase.

The non-terminal and terminal vocabulary of the language are written as:

$$V_N = \{ S_0, \text{Character, Element, Membership, PrimitiveSet, TermSet, Term, Primitive} \},$$

$$V_T = \{ \{ \text{Features (VSL, \ldots, LPy) (See Table3)} \}, \{ \text{Attributes(Z, \ldots, E)(See Table2)} \}, \{ \text{Operators(' + ', 'T', 'E')} \}, \{ \text{Modifiers('['}, ',', '<', '>', '])'} \}.$$  

A FOHDEL production rule $P$ can be written as follows:

$$S_0 \to \{ \text{Character} \to \{ \text{Element} ([\{' + '|'']|'\&']\text{Element}) \}.$$  

#### Membership

$$\text{Element} \to \{ \text{Primitive}|\text{ModifierSet # Primitive} \}.$$  

#### Membership

$$\text{ModifierSet} \to \{ \text{Term} ([\{'|']\text{Term})|'\{'|']\text{Term})|'\{'|']\text{Term}) \}.$$  

#### Membership

$$\text{Term} = \{ \text{Character}, \text{Element}, \text{Character}, \text{Element}, \text{Character}, \text{Element}, \text{Character}, \text{Element} \}.$$  

$$\text{Term} = \{ \text{Character}, \text{Element}, \text{Character}, \text{Element}, \text{Character}, \text{Element}, \text{Character}, \text{Element} \}.$$  

FOHDEL production rules $P$ are a list of primitives related to attributes through operators. The brackets $\{ \ldots \}$ denotes the optional extension of the rule. Each attributive extension contains a membership $\mu$ in the universe of discourse $[0,1]$.

Example 1. A fuzzy rule for “b” in FOHDEL according to $L(G_F)$ from Definition 2 is described through the following structure (Fig. 1), C1–C6 are the Non-terminal Elements and D is the final combination of the elements, which completes the description:

- $V_N = \{ S, C1, C2, C3, C4, C5, C6, D \}$
- $V_T = \{ \{ \text{VL}, \text{L}, \text{O}, \text{BR}, \text{PEN}, \text{THIN} \}, \{ \text{Z, L, H, VH, VVH} \}, \{ \text{>, #, and, |} \} \}$
- $S \to D \{ \mu_S = \mu_D \}$
- $C1 \to \text{VL} \text{L} \{ \mu_{C1} = \mu_{VL} = \prod_a (F[\text{VL}]) \text{Width}, \text{VH} \}$
- $C2 \to \text{H} \text{O} \text{BR} \{ \mu_{C2} = \mu_{O} = \prod_b (F[\text{O}]), \text{Width}, \text{H} \}$
- $C3 \to \text{Z} \text{PEN} \{ \mu_{C3} = \mu_{Z} \text{(\mu_{PEN})} = \prod_c (F[\text{PEN}]), \text{Width}, \text{Z} \}$
- $C4 \to \text{H} \text{O} \text{BR} \text{D} \{ \mu_{C4} = \mu_{MT}(\text{H}, \mu_{DB}) \}
- C5 \to \text{THIN} \{ \mu_{C5} = \mu_{\text{THIN}} \}$
- $C6 \to C2|C4 \{ \mu_{C6} = \mu_{\text{MAX}}(\mu_{C2}, \mu_{C4}) \}$
- $D \to C1 \text{C2} \text{C6} \{ \mu_D = \mu_{\text{MAX}}(\mu_{C1}, \mu_{C2}, \mu_{C6}) \}$

![Fig. 1. Rule generation of “b”](image-url)
The rule for a character from the above syntactic description is written as

\[
\text{Rule b: } \text{VH} \# \text{VL-L } \& \left( (\text{H} \# \text{O-BR}) > \text{H} \# \text{D-BR} \right) \\
\& \text{Z} \# \text{PEN} \& \text{THIN}
\]

The presented example shows that the main characteristics of a FOHDEL sentence are the set of features, the set of attributes related to these features, and the set of operators available to combine these features. The pattern features can be classified into global e.g. thin shaped (THIN), number of pen-ups (PEN); positional features e.g. Left (L), Bottom-Right (BR); and the geometrical features e.g. Vertical-Straight-Line (VSL), O-Like (OL).

The rule generation process in the on-line character recognition system FOHRES [16] was initialized with 250 feature primitives. The global features are basically used to reduce the classification time by supporting a hierarchical discrimination scheme [24]. The basic geometrical feature primitives are aggregated with positional and form features to create complex features e.g. Vertical-Line-to-the-Left (VL-L), O-Like-to-the-Right (O-BR). The feature aggregation process not only reduces the number of features but also increases the discrimination power of the classifier. As an example just by combining the geometrical feature Vertical-Line with the position Left or Right and the geometrical feature O-like with the position Left, Right, Top, Bottom characters like ‘p’, ‘b’, ‘q’, ‘d’ can be described uniquely, e.g. a rule for a character ‘p’ corresponds to a “Vertical-Line-Left and O-Like-Top-Right”. Through the application of aggregation mechanism to our latin character recognition system FOHRES we merged the 250 basic features into a fixed number of 120 complex features [26] (Table 3).

The next step of FOHDEL rule-base generation is the selection of linguistic attributes for the extracted features. These fuzzy feature attributes introduce additional flexibility into the feature space. They assign the qualitative measure of extracted features, i.e. how “good” or “bad” a feature (global, geometrical, complex) has to be of a certain expected value for distinguishing a particular character (Fig. 1). Thus for example for the character “b” the Vertical-Line has to be very good i.e. of high discriminatory value and hence attribute = Very-High (VH). The general qualitative features space is divided into nine linguistic terms Fig. 2. To enhance the linguistic attributes additional fuzzy hedges or modifiers e.g. “More Than Very High” (>VH), “Between < attribute X >” and < attribute Y > “ (M || H) (see Table 1) are supported.

(a) For details of membership functions see Appendix A
(b) For a list of all features see Appendix B, Table 3.

The last requirement for constructing the FOHDEL rule is to select the appropriate operators. These can be of pessimistic type like “&” (“and”; MIN) case in which the rule is valid only if all features are detected and the overall evaluation is related to the worse fit value, or the operators can be of optimistic type “|” (“or”; MAX) case in which just one fit of the possible features activates the rule with the best fit value. Other combinations between these two extremes are also supported i.e. weighted average.

4. Automatic generation of the FOHDEL rule-base

The advantage of the proposed language is not only limited to its compactness and readability but also because of its generative power and high speed in classification. This can be illustrated with the application of FOHDEL in automatic generation and modification of FOHDEL rule base with changing handwriting variability. The automatic rule generation schemes are discussed in Refs. [16,24,30]. We will insist here only on the aspects related to the language itself and its support to automatic rule generation. The primary goal of such an automatic rule-base generation system is to create an optimal number of effective rules by utilizing a minimum amount of input information. In other words the best features and the best rules should be selected and the rest of the unnecessary information should be discarded. Very frequently structure identification methods are used to extract the knowledge automatically from the raw information data, and subsequently these are converted into a special set of rules.

Our first objective in automatizing the rule generation is to retain the “good” features and discard the redundant ones. The criterion of a “good” feature is that it should be invariant to writing style that means give a good possibility answer within a class, and have a highly discriminating character related to other features. The discrimination factor can be enhanced through the linguistic attributes. The second objective is to create an automatic mechanism which can change the rule-base and thus adapt to new writers and handwriting styles. Finally the automatic rule generation method should insure the consistency of the generated rule-base.

We choose a hybrid approach that combines the knowledge of an expert with statistically generated fuzzy rules due to the time and computer power constraints required by our application [24]. The statistical information was extracted from UNIPEN database [31] which offers five million characters of various handwriting styles. The statistical information of these features is in form of the fuzzy feature space. The target of this step is to distinguish the qualitatively superior features with a positive or negative discriminatory power. The selection of good features is accomplished by histogram analysis.

---

4 Feature generation process is not elaborated here. Please refer to Ref. [29] for details of the same.
comprising of fuzzy average, fuzzy variance and fuzzy correlation matrix [24]. A fuzzy average measure provides a global feature distribution from the histogram analysis and on the other hand the fuzzy variance (σ) determines the redundancy of a particular feature. Hence the features with extremely high standard deviation are discarded because of their low discrimination power. Similarly the low-average features are also discarded unless it is manually specified by the expert to negate certain features.

For proper utilization of an extracted feature a corresponding attribute is very important. Because mere presence of a feature in a pattern is not enough, a quality measure of extracted feature is represented by the linguistic attributes, e.g. “A character b has an excellent vertical line”. This is especially useful for ambiguous situations e.g. when the training data is a compilation of various handwriting styles [31].

The initial rule base is generated from the extracted features. This rule base uses ‘&’ (AND) operator to combine different feature elements. Through an iterative process using the additional operators such as ‘||’ (Between), ‘>’ (More Than) and ‘<’ (Less Than) the rules with the neighboring attributes are merged to reduce the redundant information (see Appendix. C). Very often a particular feature has different attributes for the same symbol. Without attributes this feature could be discarded as partially relevant or irrelevant. Another aspect is the smooth variation from one symbol to a related neighbor, as shown in Figs. 2 and 3. The attributes defines the border between the different features.

Algorithm Automatic rule generation

Step 1: Acquire a prototype feature set of a large handwriting data.
Step 2: Compute the statistical parameters. Set the thresholds for statistical parameters.
Step 3: Discard the redundant features by histogram analysis and correlation.
Step 4: Determine the FOHDEL features and linguistic terms.
Step 5: Construct the FOHDEL rules.

(a) Automatic: Construct the FOHDEL rule-base with the extracted primitives and linguistic terms.
(b) Manual Correction: Edit the rule-base by removing unnecessary rules and primitives.

Fig. 2. Selection of the linguistic attributes.

Fig. 3. Variation of ‘a’ to ‘u’.
features. What ranges of values should $A_L$ and $U_L$ have to be able to discriminate ‘u’ from ‘a’. By the automatic extraction process based on the proposed fuzzy variance measure the features are extracted with their corresponding attributes and converted into FOHDEL rules. This effect was also confirmed by the test results that showed an improvement in recognition rate by 3%. The summary of the above steps for automatic rule generation is given in the Algorithm: Automatic rule generation.

The maximum frequency interval or fuzzy average peak is chosen as the central value of the corresponding linguistic term. The width of the membership function of this linguistic term is partially dependent on the fuzzy variance. If multiple peaks are in the same range of values, the linguistic term is enhanced by enlarging its boundary with help of the operator “between”, “more than” or “less than” (see Figs. 4 and 5). If peaks are not adjacent then the enlargement of the attribute is achieved through the operator “or” which reflect alternative solutions. Through this attribute aggregation the number of generated rules is reduced and problems related to the selection between several features with high variance that correspond to alternative prototypes, are solved.

5. Experimental results

After the rule generation phase in form of fuzzy rule base, recognition of an unknown handwriting sample can be undertaken. The FOHDEL rule-base is first parsed by an LALR-2 parser. In order to reduce the number of inferences needed to recognize a symbol, the prototype rules in the fuzzy knowledge base are sorted in the symbol categories. The global FOHDEL features pre-classify the unknown symbols into categories e.g. with pen-ups, wide characters, thin characters, complex characters (many specific features), simple characters (few specific features), etc. Each such category consists of several symbols and for each symbol is described by one or more FOHDEL rules. This permits a pre-classification based

---

5 The rule base from Appendix C is used for results.
on global feature evaluation and thus reduces the computation steps.

After the identification of possible categories of the unknown symbols final classification task is accomplished. For the final classification aggregated features for the unknown symbol are evaluated and a possible output membership matrix is generated. The membership values of the unknown symbol are compared with each of the selected symbol categories and the possible characters with maximum membership values are selected.

The output of the classification process is a list of prototypes with their corresponding membership value. As the estimation of the character to belong to different features is not normalized to 100%, answers of the following types are possible: 87% to belong to “a” and 60% to “u” (see Figs. 2 and 4). This is the correct answer and reflects the following situation: when the circle in the letter “a” is not closed and the recognition of the isolated character is ambiguous. The more the circle opens the more the possibility to be an “a” decreases and the possibility to be an “u” increases. This ambiguity can be solved either by choosing a (using a winner-take-it-all approach) or by adding a second-context-dependent parser which decides the final solution from the given short list of possible characters with corresponding membership values.

The real-time on-line handwriting was obtained on various platforms. The character recognition time on SPARC-2 platforms was between 0.2 and 0.6 s which is sufficient for real-time operation on personal digital assistants. The on-line performance of FOHRES for recognizing isolated handwritten characters by various users was better than other experimented methods. In Table 1 we summarized a comparison of FOHRES [32] with a neural network back-propagation algorithm (NN-BP) introduced in Ref. [17]. This performance evaluation is only qualitative and based on on-line recognition operation. Therefore for establishing a quantitative performance evaluation of FOHRES the initial set of UNIPEN benchmark data were selected. This set consisted of data from altogether ten different writers and 100 character sets of 36 symbols (Lower case latin alphabet and numerals). The classification results of this UNIPEN test data set (100 × 36 × 10 = 36,000 symbols) are shown in Table 2. The training set was not included in the test set and amounted to less than 10% of the test set. FOHRES provides a membership matrix of the five best matches. Some of the falsely classified characters are shown in Table 3. The first column shows the input symbol, the second and third column show the first and second priorities of recognizer. The last column shows the intended character. The experiments have shown that in

<table>
<thead>
<tr>
<th>Method</th>
<th>Recognition rate with unipen data*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1st Possibility</td>
</tr>
<tr>
<td>NN-BP [17]</td>
<td>94.3%</td>
</tr>
<tr>
<td>FOHRES [32]b</td>
<td>93.7%</td>
</tr>
</tbody>
</table>

*aftp://sequoyah.ncsl.nist.edu/outgoing/unipen/train_r01_v05/train_r01_v05.tar.Z
bUsing the proposed FOHDEL language.

<table>
<thead>
<tr>
<th>Input symbol</th>
<th>First recognition choice</th>
<th>Second recognition choice</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>76%</td>
<td>e</td>
<td>e</td>
</tr>
<tr>
<td>y</td>
<td>75%</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>b</td>
<td>80%</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>p</td>
<td>34%</td>
<td>h</td>
<td>b</td>
</tr>
<tr>
<td>w</td>
<td>56%</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>w</td>
<td>46%</td>
<td>n</td>
<td>m</td>
</tr>
<tr>
<td>a</td>
<td>78%</td>
<td>None</td>
<td>q</td>
</tr>
<tr>
<td>f</td>
<td>86%</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>

Table 3
False classified unknown symbols

<table>
<thead>
<tr>
<th>Input symbol</th>
<th>First recognition choice</th>
<th>Second recognition choice</th>
<th>Ground Truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>l</td>
<td>76%</td>
<td>e</td>
<td>e</td>
</tr>
<tr>
<td>y</td>
<td>75%</td>
<td>g</td>
<td>g</td>
</tr>
<tr>
<td>b</td>
<td>80%</td>
<td>h</td>
<td>h</td>
</tr>
<tr>
<td>p</td>
<td>34%</td>
<td>h</td>
<td>b</td>
</tr>
<tr>
<td>w</td>
<td>56%</td>
<td>u</td>
<td>u</td>
</tr>
<tr>
<td>w</td>
<td>46%</td>
<td>n</td>
<td>m</td>
</tr>
<tr>
<td>a</td>
<td>78%</td>
<td>None</td>
<td>q</td>
</tr>
<tr>
<td>f</td>
<td>86%</td>
<td>t</td>
<td>t</td>
</tr>
</tbody>
</table>
case of given context information the recognition rate was considerably improved by the given choice of characters with the help of fuzzy logic.

6. Conclusions

We have presented the requirements and constraints that have influenced the development of the fuzzy pattern description language FOHDEL. Through the introduced attributed features and various operators not only a manual rule-base generation as well as the automatic rule generation was possible. The power of the presented language lies not only in the simple syntax and the large possibility of attributes, features and operator combination but also in the possibility to fine tune uncertainty. The next step of our research work is to apply the proposed language to describe and recognize the cursive handwritten words. For this purpose some features have to be modified to suit the word lengths and to accommodate the repetitive features to give as in the case of the attributes a relative position, between the identified discriminatory features.

Appendix A. \( S \)- and \( \bar{S} \)- membership functions

The name \( S \)-function refers to its strong resemblance to the character ‘S’. (Fig. 8, Eq. (6)).

\[
S(x; a, b, c) = \begin{cases} 
0 & \text{for } x \leq a, \\
2 \cdot \left( \frac{x - a}{c - a} \right)^2 & \text{for } a \leq x \leq b, \\
1 - 2 \cdot \left( \frac{x - c}{c - a} \right)^2 & \text{for } b \leq x \leq c, \\
1 & \text{for } x \geq c.
\end{cases}
\]  

Similarly the \( \bar{S} \)-function is defined in terms of the \( S \)-function (Fig. 8, Eq. (7)). By putting two \( S \)-functions back to back a \( \bar{S} \)-function is obtained.

![Fig. 6. Or (\( \bar{S} \)) Operator (\( \mu_{\text{MAX}} \)).](image)

![Fig. 7. AND Operator (\( \mu_{\text{MIN}} \)).](image)
Fig. 8. S-and $\prod$ functions.

<table>
<thead>
<tr>
<th>Operators</th>
<th>Meaning</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&amp;$</td>
<td>‘and’ of two primitives</td>
<td>$\mu_{MIN}(a, b) = \frac{a + b -</td>
</tr>
<tr>
<td>$</td>
<td>$</td>
<td>‘or’ of two primitives</td>
</tr>
<tr>
<td>+</td>
<td>‘average’ of primitives</td>
<td>$\mu_{Average}(a, w_a, b, w_b) = \frac{a \cdot w_a + b \cdot w_b}{w_a + w_b}$</td>
</tr>
<tr>
<td>$&gt;$</td>
<td>more than a linguistic attribute</td>
<td>$\mu_{M}(a, x) = \begin{cases} S(x, a - 0.5, a - 0.25, a) &amp; \text{for } x \leq a, \ 1 &amp; \text{otherwise.} \end{cases}$ (Fig. 4a)</td>
</tr>
<tr>
<td>$&lt;$</td>
<td>less than a linguistic attribute</td>
<td>$\mu_{L}(a, x) = \begin{cases} S(x, a, a - 0.25, a - 0.5) &amp; \text{for } x \leq a, \ 0 &amp; \text{otherwise.} \end{cases}$ (Fig. 4b)</td>
</tr>
<tr>
<td>$|$</td>
<td>between two linguistic attributes</td>
<td>$\mu_{BET}(a, b, x) = \begin{cases} S(x, a - 0.5, a - 0.2, a) &amp; \text{for } x \leq a, \ 1 &amp; \text{for } a \leq x \leq b, \text{ (Fig. 5)} \ S(x, b - 0.5, b - 0.25, b - 0.5) &amp; \text{for } x \geq c \end{cases}$</td>
</tr>
<tr>
<td>#</td>
<td>Separator between a linguistic term and the primitive</td>
<td></td>
</tr>
<tr>
<td>( )</td>
<td>Brackets $s$ are used to differentiate the hierarchies as in case of $+$ operators</td>
<td></td>
</tr>
</tbody>
</table>

\[
\prod(x; b, c) = \begin{cases} S(x; c - b, c - \frac{b}{2}, c) & \text{for } x \leq c, \\ 1 - S(x; c, c + \frac{b}{2}, c + b) & \text{for } x \leq c. \end{cases}
\] (7)

In $S(x; a, b, c)$ the parameter $b, b = (a + c)/2$, is a crossover point. A crossover point is a value of $x$ at which $S$ attains equilibrium, i.e. 0.5. In $\prod(x; b, c) b$ is the band-width, i.e. distance between the two crossover points of $\prod$. At the center height of $\prod$ is unity. This demonstrates that $\prod$ represents normal fuzzy functions.

Appendix B

Elements of FOHDEL language are in Tables 4–6.
Table 5
FOHDEL linguistic attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Symbol</th>
<th>Parameters</th>
<th>Membership Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero</td>
<td>Z</td>
<td>$b = 0.30, c = 0.00$</td>
<td>$\mu_Z(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Very Very Low</td>
<td>VVL</td>
<td>$b = 0.30, c = 0.15$</td>
<td>$\mu_{VVL}(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Very Low</td>
<td>VL</td>
<td>$b = 0.30, c = 0.30$</td>
<td>$\mu_{VL}(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Low</td>
<td>L</td>
<td>$b = 0.30, c = 0.40$</td>
<td>$\mu_L(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>$b = 0.30, c = 0.50$</td>
<td>$\mu_M(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>High</td>
<td>H</td>
<td>$b = 0.30, c = 0.60$</td>
<td>$\mu_H(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Very High</td>
<td>VH</td>
<td>$b = 0.30, c = 0.70$</td>
<td>$\mu_{VH}(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Very Very High</td>
<td>VV H</td>
<td>$b = 0.30, c = 0.85$</td>
<td>$\mu_{VV H}(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
<tr>
<td>Excellent</td>
<td>E</td>
<td>$b = 0.30, c = 1.00$</td>
<td>$\mu_E(x) = \left{ \begin{array}{ll} 1 &amp; 0 \leq x &lt; 0.30 \ 0 &amp; \text{otherwise} \end{array} \right.$</td>
</tr>
</tbody>
</table>

Table 6
FOHDEL features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSL</td>
<td>Vertical Straight Line</td>
</tr>
<tr>
<td>PS</td>
<td>Positive Slant</td>
</tr>
<tr>
<td>NS</td>
<td>Negative Slant</td>
</tr>
<tr>
<td>O.L</td>
<td>O-Like Curve</td>
</tr>
<tr>
<td>C.L</td>
<td>C-Like Curve</td>
</tr>
<tr>
<td>D.L</td>
<td>D-Like Curve</td>
</tr>
<tr>
<td>A.L</td>
<td>A-Like Curve</td>
</tr>
<tr>
<td>U.L</td>
<td>U-Like Curve</td>
</tr>
<tr>
<td>HL.T</td>
<td>Horizontal Line on the Top</td>
</tr>
<tr>
<td>HMN</td>
<td>Horizontal Motion</td>
</tr>
<tr>
<td>HL.M</td>
<td>Horizontal Line on the Middle</td>
</tr>
<tr>
<td>VMN</td>
<td>Vertical Motion</td>
</tr>
<tr>
<td>HL.B</td>
<td>Horizontal Line on the Bottom</td>
</tr>
<tr>
<td>PS.TL</td>
<td>Positive Slant on the Top-Left</td>
</tr>
<tr>
<td>PS.MM</td>
<td>Positive Slant on the Middle-Middle</td>
</tr>
<tr>
<td>PS.BM</td>
<td>Positive Slant on the Bottom-Middle</td>
</tr>
<tr>
<td>NS.TM</td>
<td>Negative Slant on the Top-Left</td>
</tr>
<tr>
<td>NS.MM</td>
<td>Negative Slant on the Middle-Middle</td>
</tr>
<tr>
<td>NS.BM</td>
<td>Negative Slant on the Bottom-Middle</td>
</tr>
<tr>
<td>O.L.T</td>
<td>O-Like Curve on the Top-Left</td>
</tr>
<tr>
<td>O.MM</td>
<td>O-Like Curve on the Middle-Middle</td>
</tr>
<tr>
<td>O.BM</td>
<td>O-Like Curve on the Bottom-Middle</td>
</tr>
<tr>
<td>D.L.T</td>
<td>D-Like Curve on the Top-Left</td>
</tr>
<tr>
<td>D.MM</td>
<td>D-Like Curve on the Middle-Middle</td>
</tr>
<tr>
<td>D.BM</td>
<td>D-Like Curve on the Bottom-Middle</td>
</tr>
<tr>
<td>C.L.T</td>
<td>C-Like Curve on the Top-Left</td>
</tr>
<tr>
<td>C.MM</td>
<td>C-Like Curve on the Middle-Middle</td>
</tr>
<tr>
<td>C.MR</td>
<td>C-Like Curve on the Middle-Right</td>
</tr>
</tbody>
</table>

Walking Stick tilted to the Left
Walking Stick tilted to the Right
Hockey stick tilted to the Left
Hockey stick tilted to the Right
Vertical Line on the Left
Vertical Line on the Middle
Vertical Line on the Right
Table 6 (Continued)

C.BL C-Like Curve on the Bottom-Left C.BM C-Like Curve on the Bottom-Middle C.BR C-Like Curve on the Bottom-Right
A.TR A-Like Curve on the Top-Left A.TM A-Like Curve on the Top-Middle A.TR A-Like Curve on the Top-Right
A.ML A-Like Curve on the Middle-Left A.MM A-Like Curve on the Middle-Middle A.MR A-Like Curve on the Middle-Right
A.BL A-Like Curve on the Bottom-Left A.BM A-Like Curve on the Bottom-Middle A.BR A-Like Curve on the Bottom-Right
U.TL U-Like Curve on the Top-Left U.TM U-Like Curve on the Top-Middle U.TR U-Like Curve on the Top-Right
U.ML U-Like Curve on the Middle-Left U.MM U-Like Curve on the Middle-Middle U.MR U-Like Curve on the Middle-Right
U.BL U-Like Curve on the Bottom-Left U.BM U-Like Curve on the Bottom-Middle U.BR U-Like Curve on the Bottom-Right
HOL.L Left sided Hockey stick on the Left HOL.R Left sided Hockey stick on the Right S.X/Y Starting X/Y Coordinate
HOR.L Right sided Hockey stick on the Left HOR.R Right sided Hockey stick on the Right E.X/Y End X/Y Coordinate
STL.L Left sided Walking Stick on the Left STL.R Left sided Walking Stick on the Right PEN Number of Penups
STR.L Right sided Walking Stick on the Left STR.R Right sided Walking Stick on the Right SEG Number of handwriting Segments
LPb Loop as in ‘b’ (Upwards) AR Aspect Ratio LPy Loop as in ‘y’ (Downwards)

Appendix C

Example set FOHDEL rules
Rule a1: ( ( > M#O-MM) | ( > M#O-TM) ) & (HOR-R | VL-R | C_BR | NS_MR | NS-MM) & ( < L#S_Y) & ( > M#S_X)
Rule a2: ( < L#S_Y) & ( > M#O-ML) | ( > H#C-MM) ) & (HOR-R | VL-R | C_MR | NS-MM | NS-MR)
Rule a3: ( > M#O-MM) & (HOR-R | VL-R | VL-RC-MR) & ( > H#S_Y) & ( > VH#S_X) & Z#VL-R
Rule b1: VL-L & (D_MR | D_BM | O_MR | O_BM | O_BR)
Rule b2: VL-L & M#LPh & (HOR-M | C_MR | C-MM) & VVL#SEG
Rule b3: VL-L & (A BM (A_BR) & L#SEG & (H#NS-BM | HSL)
Rule b4: VL-M & O_BM
Rule b5: PS-ML & C-MM & (HL-M | U-MR)
Rule c1: C-MM & ( < VL#SEG) & ( < L#HSL)
Rule c2: C-MM & ( < VL#SEG)
Rule d1: (C_ML | C-MM | 0 MM) & (HOR-R | VL-R) & ( > M#S_Y) & ( < VH#S_X) & ( < L#HSL)
Rule d2: STR & VL-R & PS-MM
Rule d3: ( > M#O-MM) | ( > M#O-TM) & (HOR-R | VL-R | C_BR | NS_MR | NS-MM) & ( > L#S_Y) & ( < H#S_X)
Rule e1: Z#VSL & Z#HSL & Z#PS & Z#O-L & E#E.X & E#E.Y & ( < M#S_X) & ( < L#S_Y)
Rule e2: ( > M#HL M) | ( > M#PS-MM)) & C_L & VL#SEG & ( < VH#O_L)
Rule f1: LPb & STR
Rule f2: LPb & LPy & HL_M
Rule f3: Z#PEN & Z#O-L & ((VH || VH))#VL-M & ((M)[H#AR)
Rule f4: Z#PEN & VH#C-L & VH#LPb & H#AR
Rule g1: (O_TL | O_TM) & ( > VH#LPy) & ( < M#PS)
Rule g2: (O_TL | O_TM) & M# VL-M
Rule h1: VL_L & (A_BM | A_BR | A MM) & ( > H#E_X)
Rule h2: PS-MM & E#E_X & Z#VSL & HSL
Rule h3: Z#PEN & ((VH || VH))#LPb & ((VH || VH)#A-L) & L#SEG
Rule i1: HOR & ( < VL#SEG)
Rule i2: HOR-R & PS ML
Rule j1: LPy & ( < VL#SEG) & ( < VH#AR)
Rule j2: LPy & PS-ML & Z#HSL
Rule k1: H#PS & H#PS & L#D_L & (VH || VH)#HOR & VH#SEG
Rule k2: (VH || VH)#STR & VH#SEG & VL_A-L
Rule l1: LPb & ( < VL#SEG)
Rule l2: ( > VH#VL-R) & ( > H#VL-M) | ( > H#PS) & ( < H#AR)
Rule l3: VL>L & (H#Lpb) & (VL#HSL) & VL#SEG
Rule m1: AML & A-MM & AMR & (VL#SEG)
Rule n1: (ML | AMR) & <M#MM & NS-ML
Rule n2: STL & A L & VVL#SEG
Rule o1: O-MM & HL-T
Rule o2: O-MM & (L#SEG)
Rule p1: VL L & (D-MM | O-MM | D-TM)
Rule p2: VL L & Lpb & (H#D-MM) & (H#D TR) & (H#D-MR)
Rule q1: O- TL & VL-M | VL-L & (PS BM | PS- BR)
Rule q2: VL M & (C-MM | O-MM | O-TM | C-TM) & PS- BR
Rule r1: VL L & HL T & (PS- ML | VL M)
Rule r2: VL L & STR
Rule s1: (> S X) & (< S Y) & (< L #E-X) & (> VH #E-Y) & VVL#SEG
Rule t1: HTR & L#PEN & (HSL | PS-MM)
Rule t2: Z#OL & ((U ML & (HOR | (< VL#SEG)) & (U MM & Z#SEG)) & (< VL#HSL)
Rule u1: Z#O L & (VH C ML | VH UML) & (H#NS & (> VH #E-Y))
Rule v1: U BL & > H#HL T & (< L#SEG)
Rule w1: U-ML & U MR & (< L#SEG)
Rule x1: (M#PS) & (> M#NS) & (Z#PEN)
Rule x2: D ML & C-MR & (< Z#SEG)
Rule y1: Z#PEN & (U-TM | U TL) & (> VH#Ppy)
Rule z1: Z#PEN & VVL#SEG & (< VL#S X) & (< VL#S Y) & (> VJ #E-X) & (> VH#E-Y)

References


About the Author—ASHUTOSH MALAVIYA is currently Chief Technology Officer at Yatra Corporation in Minneapolis. He has worked as a scientist at the German National Research Center for Information Technology – GMD for the past six years. Previously he worked at the Fraunhofer Institute for Production and Construction Engineering in Berlin, Germany. His main research interests include fuzzy pattern recognition, computational intelligence and decision support systems. Dr. Malaviya obtained the Bachelors degree in Electrical Engineering from the University of Roorkee, India. He obtained the Master’s degree in Information Engineering and a Ph.D. in Computer Science from the Technical University of Berlin, Germany.

About the Author—LILIANE PETERS is a senior scientist at GMD-SET. Previously she worked as a Research Assistant in the Department of Electrical Engineering at the Technical University in Aachen, Germany. Her main interests are fuzzy systems and real-time aspects of computer vision. Dr. Peters received the Dipl.-Ing. degree in Electrical Engineering from the Technical University in Bucharest and the Ph.D. degree from the Technical University in Aachen, Germany.