Reprint: 1995 Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP-95), Detroit, USA, Vol. 4, pp. 2459-2462, May 1995

A SOFT-DECISION APPROACH FOR STRUCTURAL ANALYSIS OF HANDWRITTEN MATHEMATICAL EXPRESSIONS

H.-J. Winkler, H. Fahrner, M. Lang
Institute for Human-Machine-Communication
University of Technology, Munich
Arcisstr. 21, 80333 Munich, Germany

ABSTRACT

In this paper an efficient system for structural analysis of handwritten mathematical expressions is proposed. To handle the problems caused by handwriting, this system is based on a soft-decision approach. This means that alternatives for the solution are generated during the analysis process if the relation between two symbols within the expression is ambiguous. Finally a string containing the mathematical information is generated and syntactical verified for each alternative. Strings failing this verification are considered as invalid.

1 INTRODUCTION

We are accustomed to writing mathematical expressions containing integrals, fractions, exponents or indices by hand. Entering these expressions into a computer is quite uncomfortable and expendable because we have to learn a notation such as TEX or we should be familiar with a graphical formula-editor supplied with mouse and keyboard [1]. The human-adapted way is analysing the handwritten expressions. But there are two essential problems to be solved, namely symbol recognition and structure analysis.

Remarkable research activities are concerned with recognizing printed or handwritten symbols, hence we are not going to focus on these problems in the following [2][3].
The structure analysis process is driven by the results obtained during symbol recognition. The input data consist of the symbol codes and the coordinates of the surrounding rectangle to each symbol [4].
At first, the stages of the analysis process are described by analysing a printed expression (hard-decision system).
After that, we discuss the problems arising in analysing handwritten expressions. In order to handle these problems, we changed the above mentioned hard-decision system into our soft-decision system by implementing some powerful extensions.
In the last section the effectiveness of our system is illustrated by analysing handwritten and printed expressions.

2 ANALYSING PRINTED EXPRESSIONS

The structure of each mathematical expression is describable by a directed graph [5]. The nodes of the graph represent the symbols, the edges describe the relations between the symbols. Five kinds of edges (*lin*, *up*, *lo*, *exp* and *ind*) are necessary.

The process of generating a directed graph by analysing the input data is illustrated in the first three stages of the block diagram shown in fig. 1 [6]. In the last stage, a string containing the mathematical information is generated and verified syntactically.

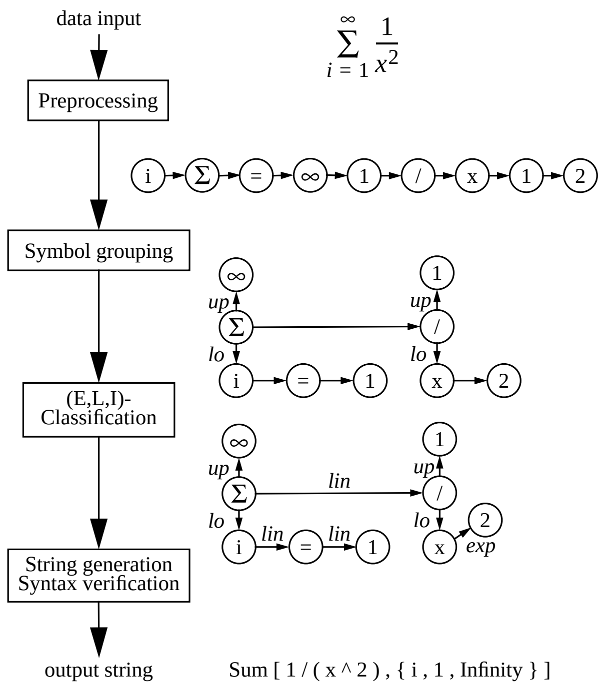

Figure 1: System overview and example

Reprint: 1995 Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP-95), Detroit, USA, Vol. 4, pp. 2459-2462, May 1995

2.1 Preprocessing

The input data (symbol code and position) are sorted from left to right. This sorted list is transformed into a directed graph in which the type of the edges is not defined yet (fig. 1).

2.2 Symbol grouping

A symbol group is defined as a special mathematical symbol including symbols which appear with it [6]. The position of these accompanying symbols depends on their meaning relative to the special mathematical symbol. For example, the fraction symbol separates the according symbols into symbols belonging to the numerator and symbols belonging to the denominator. Each symbol group itself can be a member of a larger symbol group.
Our system currently regards five special symbols: fraction (numerator and denominator); summation, product and integration (upper and lower limits); root (radicand and power). At first, the system searches for special mathematical symbols within the graph. For each located special symbol a grouping process is started next. Thereby symbols, whose position is above or below the considered special symbol, are detected. For the special symbol „root“ symbols belonging to the radicand or to the power of the root are searched. These symbols are sorted again and connected to the special symbol using the edges *up* or *lo* (depending on the position) within the directed graph (fig. 1).

This grouping process is a hard-decision process, which means for example, that a symbol belongs to the denominator of a fraction symbol if its surrounding rectangle is completely below the surrounding rectangle of the fraction symbol [4].

2.3 (<u>E</u>xponent,<u>L</u>ine,<u>I</u>ndex)-classification

This stage defines the remaining undefined edges of the graph. There are three possible relations between two symbols connected by an undefined edge: the second symbol is either on the same line as the first symbol (*lin*) or it is part of the exponent (*exp*) or the index (*ind*) of the first symbol. The classification into one out of these three classes is done by analysing the relative positions of the surrounding rectangles of these two symbols [5].

2.4 String generation and syntactical verification

Based on the resulting directed graph a string containing the two-dimensional information from the input data is generated. The coding of the two-dimensional information is done by using the syntax of MATHEMATICA [7]. After generating the string a syntactical verification is done by means of MATHEMATICA.

3 ANALYSING HANDWRITTEN EXPRESSIONS

In comparison to analysing printed expressions, the analysis of handwritten mathematical expressions raises additional problems caused by variations in the positioning of the symbols within the expression, variations of the symbol size, writer-dependent slant (horizontal and/or vertical), etc.
Fig. 2 illustrates these problems by an example, which might be part of a larger mathematical expression.

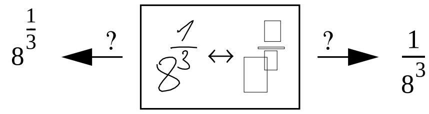

Figure 2: Ambiguous handwritten expression and the corresponding interpretation possibilities

Obviously, in this case even we have difficulties in analysing the expression. We might be able to do an unambiguous classification if further knowledge such as the writing line or additional context information is available.

In order to handle such ambiguous elements, the system is changed into a soft-decision system, the analysing strategy illustrated in fig. 1 however stays unchanged.

For this soft-decision approach in analysing (handwritten) mathematical expressions, modifications are necessary in the following stages:

3.1 Symbol grouping

The surrounding area (not identical with the surrounding rectangle) of each special symbol is split into different regions. The arrangement of these regions as well as the calculation of the corresponding probabilities described in the following depends on the kind of special symbol focused during the grouping process.

Using the fraction symbol given in fig. 3, the soft-decision grouping process is now described.

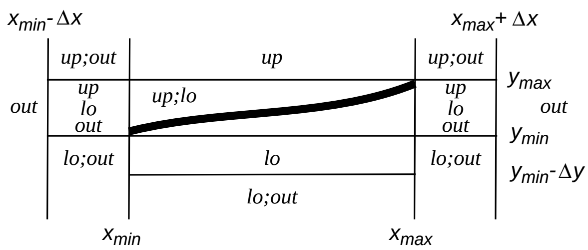

Figure 3: Surrounding area of the fraction symbol; *out* stands for outside symbol group

The distances Δx and Δy are calculated by the size of the symbols within the expression, furthermore Δx depends on the size of the fraction symbol as well as on the position of the symbols already connected to the fraction symbol by the relations *up* or *lo*.

Depending on the symbol position (sx_{min}, sx_{max}, sy_{min}, sy_{max}) in relation to the position (x_{min}, x_{max}, y_{min}, y_{max}) of the fraction symbol, three distances d_{up}, d_{lo} and d_{out} are calculated, which represent a measurement for belonging to the corresponding regions *up*, *lo* and *out*. The basic idea in calculating the distance d_i, $i \in \{up, lo, out\}$, is founded on the amount of shifting necessary for a non-ambiguous classification of the relation between the symbol and the special symbol.

If the surrounding rectangle of the symbol is completely within the single region *i*, the relation is non-ambiguous, therefore

$$d_i \to 0;\ d_j \to \infty;\ \ i, j \in \{up, lo, out\}, j \neq i\ .$$

If the symbol touches a double region (*i*;*j*), (*i*;*j*) ∈ {(*up*;*lo*), (*up*;*out*),(*lo*;*out*)}, the distances d_i and d_j are calculated, the remaining distance d_k is set to

$$d_k \to \infty;\ k \in \{up, lo, out\}, k \neq j \neq i\ .$$

All three distances are calculated for a symbol touching one of the two triple regions (*up*;*lo*;*out*). As an example, for a symbol touching the triple region placed left to the fraction symbol in fig. 3, the calculation is done by

$$d_{up} = a \cdot (x_{min} - sx_{min}) + c \cdot (y_{max} - sy_{min})\ ;$$

$$d_{lo} = a \cdot (x_{min} - sx_{min}) + c \cdot (sy_{max} - y_{min})\ ;$$

$$d_{out} = \left| (y_{max} - sy_{min}) - (sy_{max} - y_{min}) \right| .$$

The amplification factor *a* depends on the extend of overlapping of the symbols surrounding rectangle with the region *out*, the variable *c* depends on the number of preceeding symbols within the actual grouping process classified as *out*.

Based on these distances, the corresponding probabilities are calculated and normalised by

$$p_i = \frac{1/d_i}{1/d_{up} + 1/d_{lo} + 1/d_{out}}\ ,\ i \in \{up, lo, out\}$$

and finally compared with an upper and a lower probability threshold p_1 and $p_0 = (1 - p_1)/2$:

- $p_i \geq p_1\ ,\ i \in \{up, lo, out\}$:
  The relation *i* between the symbol and the special symbol is assumed to be non-ambiguous, therefore the probabilities are set to
  $$p_i = 1;\ p_j = 0;\ i, j \in \{up, lo, out\}, i \neq j\ .$$
  Within the directed graph the edge *i* is used for connecting the symbol to the special symbol.
- $p_i \leq p_0\ ,\ i \in \{up, lo, out\}$:
  The probability p_i of the relation *i* is too small in comparison to the remaining probabilities p_j, therefore the probability p_i is set to $p_i = 0$, the remaining two probabilities p_j are normalised again. For each relation *j* a directed graph is generated by doubling the currently processed graph. Within these two alternatives, different kinds of edges (corresponding to the relations *j*) are used for connecting the symbol to the special symbol and the corresponding probabilities p_j are stored.
- $p_1 > p_i > p_0\ ,\ i \in \{up, lo, out\}$:
  Three alternatives are generated by duplicating the currently processed graph, within these alternatives the edges *i* are used for connection.

This hard-decision within the soft-decision grouping process is necessary because the number of alternatives generated during the analysis process of a large mathematical expression may exceed the available memory.

The result of the soft-decision symbol grouping process is illustrated in fig. 4 using the expression given in fig. 2.

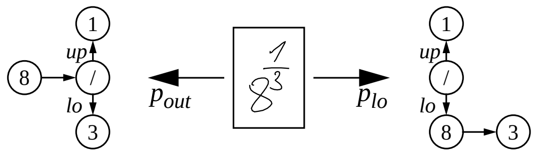

Figure 4: Generation of two alternatives caused by the symbol " 8 " in region (*lo*;*out*)

3.2 (E,L,I)-classification

The remaining undefined edges of each graph have to be classified as *lin*, *exp* or *ind*. Possible mistakes in the hard-decision process occur only between *lin* and *exp* or between *lin* and *ind*.

Therefore, three probabilities p_{lin}, p_{exp} and p_{ind}, which represent a measurement for the corresponding relation between two neighbouring symbols within the graph, are calculated by analysing

- the relative position of the surrounding rectangles.
- the relative position of the center of each symbol. The center sy_{cen} of a symbol *s* depends on the coordinates (sx_{min}, sx_{max}, sy_{min}, sy_{max}) of the surrounding rectangle and the symbol itself. The calculation of sy_{cen} is done by
  $$sy_{cen} = sy_{min} + g \cdot (sy_{max} - sy_{min})\ ,$$
  The factor *g* is determined by a categorisation of the symbols into one out of three different classes depending on the symbol code :

- symbols consisting of a center part (such as “ a “, “ c “, “ e “, “ m “, “ + “, “ - “, “ = “, ...) or
- symbols consisting of a center part as well as an upper and a lower part (such as “ f “, “ ∫ “, ...): $g = 0.5$.
- symbols consisting of a center and an upper part (such as capital letters, numbers, “ b “, “ d “, ...): g = 0.25 .
- symbols consisting of a center and a lower part (such as “ g “, “ p “, “ q “, ...): $g = 0.75$.

Fig. 5 illustrates the usefulness of this categorisation.

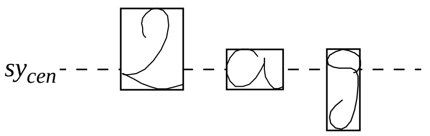

Figure 5: The handwritten symbols “ 2 a g“, aligned to their center position sy_{cen}

Analogous to the symbol grouping process, the probabilities p_{lin} , p_{exp} and p_{ind} are compared with the two thresholds p_1 and p_0 , again to limit the number of alternatives.
If the relation between the two symbols is non-ambiguous, the edge between these symbols is defined corresponding to that relation. On the other hand, if the relation is ambiguous, the graph is duplicated and different kinds of edges are used for describing the relation between these two symbols; the corresponding probabilities are stored.

3.3 String generation and syntactical verification

For each directed graph a character string is generated. These strings are sorted with regard to the probabilities obtained during the symbol grouping and the (E,L,I)-classification process. The generated strings are syntactically verified, the strings failing the verification are deleted. The remaining alternatives represent the output of the analysis process. If there is more than one alternative, the user has to choose the correct one.

4 SOME ANALYSER RESULTS

Each handwritten expression given in fig. 6 was analysed by the system. The input data are generated by using a symbol recognition system suggested in [8][9].

The best alternative offered by the system was always the correct solution. On average 9 syntactical correct alternatives are generated during the analysis process. Most of these alternatives are based on an ambiguous relation between two symbols within the (E,L,I)-classification process.

Furthermore, we tested the soft-decision system by analysing the corresponding printed expressions (typeset by $\TeX$). In this case, a scanner and a human symbol recognizer generated the input data. In analysing the printed expressions,

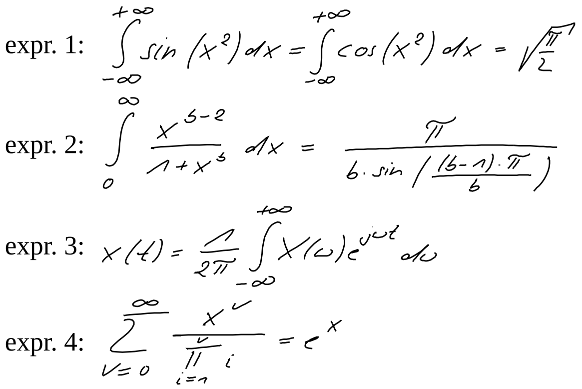

Figure 6: Some handwritten expressions used for testing

the system offered 2 alternatives on average, the correct solution always being the most probable.

5 REFERENCES

[1] D. E. Knuth, *The $\TeX$book*, Addison-Wesley, 1986.

[2] C. C. Tappert, C. Y. Suen, T. Wakahara, *The State of the Art in On-line Handwriting Recognition*, IEEE Trans. PAMI Vol.12 No.8, 1990.

[3] K. S. Nathan, J. R. Bellegarda, D. Nahamoo, E. J. Bellegarda, *On-line Handwriting Recognition using Continuous Parameter Hidden Markov Models*, ICASSP 1993 Vol.5, 1993.

[4] R. H. Anderson, *Syntax-directed recognition of hand-printed two-dimensional mathematics* in *Interactive Systems for Experimental Applied Mathematics* (M. Klerer and J. Reinfelds, ed.), Academic, 1968.

[5] Z.-X. Wang, C. Faure, *Structural analysis of handwritten mathematical expressions*, 9th Conference on Pattern Recognition, 1988.

[6] H.-J. Lee, M.-C. Lee, *Understanding mathematical expressions using procedure-oriented transformation*, Pattern Recognition Vol.27 No.3, 1994.

[7] S. Wolfram, *Mathematica: A System for Doing Mathematics by Computer*, 2nd edition, Addison-Wesley, 1991.

[8] H.-J. Winkler, *Symbol Recognition in Handwritten Mathematical Expressions*, Proc. Int. Workshop „Modern Modes of Man-Machine-Communication“, Maribor (Slovenia), 1994.

[9] M. Koschinski, H.-J. Winkler, M. Lang, *Segmentation and Recognition of Symbols within Handwritten Mathematical Expressions*, to be published in ICASSP 1995