2009 10th International Conference on Document Analysis and Recognition

Towards Handwritten Mathematical Expression Recognition

Ahmad-Montaser Awal, Harold Mouchère, Christian Viard-Gaudin
IRCCyN/IVC – UMR CNRS 6597
Ecole polytechnique de l'université de Nantes
Rue Christian Pauc – BP 50609 – 44306 Nantes CEDEX 3 – France
{ahmad-montaser.awal, harold.mouchere, Christian.Viard-Gaudin }@univ-nantes.fr

Abstract

In this paper, we propose a new framework for online handwritten mathematical expression recognition. The proposed architecture aims at handling mathematical expression recognition as a simultaneous optimization of symbol segmentation, symbol recognition, and 2D structure recognition under the restriction of a mathematical expression grammar. To achieve this goal, we consider a hypothesis generation mechanism supporting a 2D grouping of elementary strokes, a cost function defining the global likelihood of a solution, and a dynamic programming scheme giving at the end the best global solution according to a 2D grammar and a classifier. As a classifier, a neural network architecture is used; it is trained within the overall architecture allowing rejecting incorrect segmented patterns. The proposed system is trained with a set of synthetic online handwritten mathematical expressions. When tested on a set of real complex expressions, the system achieves promising results at both symbol and expression interpretation levels.

1. Introduction

For centuries, handwritings had been the most common way of communication. Nowadays, computers and the Internet are the indispensable way of modern communications; turning the world into a small town.

However, using handwritings in this communication world is still something very appreciable and which remains possible, especially with the emergence of devices that rely on digital pens as an input method. PDAs, tablet PCs or electronic white boards are such examples. Developing those devices was accompanied with the development of recognition systems capable of converting texts from our natural handwriting, into languages understandable by computers.

Handwritten text recognition systems have achieved recently a significant progress, thanks to developments in segmentation, recognition and language models. Those systems are less powerful when the languages to be recognized have a two dimensional (2D) layout. Which is the case for mathematical expressions [1], schemas, diagrams, etc. In this case, it yields to solve the same problems of segmentations, recognition and interpretation but in a 2D context.

Mathematics is widely used in almost all fields of science, such as physics, engineering, medicine, economics, etc. As a result, an input method for mathematical expressions into scientific documents is a requirement.

Many tools are available to achieve this task. However, most of those tools require some expertise to use them efficiently. Latex and MathML, for example, require knowledge of predefined sets of key words to describe special mathematical symbols and functions in addition to spatial layouts. Other tools, such as Math Type, depend on a visual environment to add symbols using the mouse and though needs lot of time.

Our research focuses on the recognition of online handwritten mathematical expressions (ME). Some researches are emerging in this area, especially on subclasses of mathematical expressions with some promising results. Most of those research works consider recognition of mathematical expressions as a set of subtasks to perform different steps of the recognition process. Though, a main drawback comes from the fact that any error at any step will be automatically inherited to the next step, requiring further processing to be sure of a good and correct recognition results.

Our contribution to the domain of online handwritten mathematical expressions recognition is to perform a simultaneous segmentation, recognition and interpretation of mathematical expressions. Specifically, the classifier used to recognize the basic symbols is based on a global learning method allowing the system to learn symbols directly from expressions instead of using a pre-trained classifier.

In section two, we introduce the problem of mathematical expression recognition. Then, we develop our own architecture in section three, and we give some preliminary results that are compared with some of other works [19].

2. Mathematical expression recognition

Being able to input a mathematical expression into a digital document might be a difficult process. Tools, such as LATEX or MathML), aim at making it an easier and more convivial process. However, complex expressions require much efforts and time. Using a digital pen presents a more natural way to input such expressions into digital documents.

Any ME recognition system must take in consideration the particularity of mathematical expressions. Those particularities arise from the huge number of symbols - compared to normal text – arranged in a two dimensional layout. Resolving ME recognition problem implies being capable of solving three sub

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.71

problems [2]: segmentation, symbol recognition and expression interpretation.

2.1. Mathematical expression segmentation

Considering an online handwritten ME signal, the primitive unit, which allows to segment it, is a stroke. A stroke being a trace drawn between a pen down and a pen lift. However, in most of the cases a single symbol is composed of several strokes. Conversely, we will assume that one pen lift exists between consecutive symbols. Hence the segmentation step will consist in grouping strokes belonging to the same symbol. Figure 1 shows possible groupings of a set of strokes. Strokes grouping problem is by itself a complex and challenging problem. This problem is sill more difficult when delayed strokes are present. We will consider those cases, and assume the eventuality of interspersed strokes between symbols.

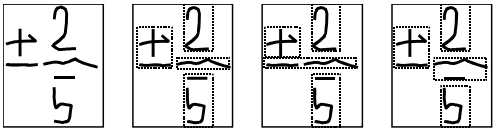

Figure 1. Example of grouping hypothesis

Faure and Wang [3] proposed two modules depending on successive X Y projections and a spatial relation tree in order to group strokes belong to the same symbol together. A similar method in [4] depends on strokes bounding boxes instead of projections. A simultaneous recognition and segmentation was introduced in [5, 19], in this method the segmentation process is lead by a recognition where the groups most likely to represent a symbol are grouped together.

After segmenting the expression we need to give each group of strokes a label identifying what it represents. This is the second step of ME recognition.

2.2. Mathematical symbol recognition

Symbol recognition problem is a more classic pattern recognition problem. However, mathematical symbol recognition is more complex regarding the huge number of symbols – more than 220 symbols are required to cover correctly most of the scientific applications. Another difficulty is the similarity among some symbol classes. For example, a horizontal segment can be recognized as a minus but can also be a fraction bar [6].

To address this problem, many classifying methods can be investigated. A template matching method is used by some systems [7, 19], however this method can be slow and time consuming. Structural recognition methods are less used in mathematical expressions recognition. Systems as those in [2, 8] extract structural primitives and use them by comparing with the training data. On the other hand, artificial neural networks (ANN) are known to be better in terms of speed and recognition rate [9, 10]. Some methods perform a simultaneous segmentation and recognition such as hidden markov model (HMM) [11], they are based on statistical models. Each symbol has its own model where recognition results are obtained as probabilities of different models.

2.3. Mathematical expression interpretation

Finally, the interpretation phase implies a structural analysis of spatial relations among symbols to find the structured description of the expression [12].

Spatial relations can cause many ambiguities that are solved lately depending on a 2D grammar that describes mathematical expressions. As shown in Figure 2, in the expression b_c the "c" can be a subscript of b as in $b_c d$, or it can be a variable as in $a^b c$, while b is a superscript of a previous variable [13].

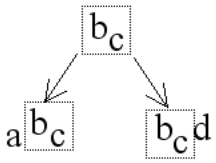

Figure 2. Local ambiguity from [13]

Spatial relationships are usually not enough to resolve all ambiguities. Thus, a syntax analysis is necessary in order to find the structure of the expression. Besides, mathematical expressions are considered as a two dimensional language. This implies the need of a 2D grammar in order to be able to analyze ME structures [14]. Achieving a 2D parsing is a complex process, which requires special techniques and methods to reduce its complexity [15].

In the proposed framework, we try limit the main drawbacks related to each step of ME recognition. We propose a framework allowing a simultaneous segmentation, recognition, and interpretation.

3. Recognition framework

The originality of the proposed framework relies on the following points. First, the symbol classifier will be trained from scratch in the context of the whole system including segmentation, 2D parsing and grammar rules to give the best possible interpretation. Different objective functions have been considered, included maximum likelihood (ML) and maximum mutual information (MMI). Second, the 2D parsing supports non-consecutive stroke grouping to be considered as a valid symbol hypothesis. Third, the 2D parsing is controlled by a combination of the symbol recognition scores and of a contextual analysis, which is specific to the domain of ME. And finally, we have defined a set of rules to define a grammar that allow checking the validity of the proposed interpretations.

We compare our results with those in [19]. They propose a layered search framework for ME recognition performing a simultaneous segmentation and recognition, but using a classifier which is trained on isolated symbols.

3.1. Global architecture

An online mathematical expression is input to the system as a set of strokes. So, recognizing an expression consists in finding the best possible grouping of those strokes to represent expression symbols and spatial relations among those symbols to find out the structure of the expression.

Both expression recognition and system training can be described with the same global architecture of expression recognizer system shown in figure 3.

It includes:

- A symbol hypothesis generator (SHG): SHG lists all the possible combination of strokes. Each group of strokes is called a symbol hypothesis (SH).
- A symbol recognizer (SR): SR provides, in addition to the label of each hypothesis, a recognition score that will be used to define the recognition cost. A multi layer perceptron (MLP) with one hidden layer is used as a recognizer with a softmax output function in order to get recognition scores as a probability.

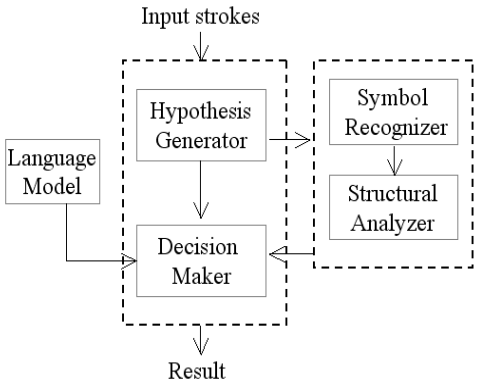

Figure 3. Expression recognizer architecture overview

- Structural analyzer (SA): SA provides structural information about each hypothesis so that contextual evaluation can be performed. It includes information such as height (h), baseline position (y_0), and strokes length of the considered hypothesis. Structural analyzer must take in consideration the particularity of each type of symbols. Figure 4 shows two different cases of structural information based on the symbol label.

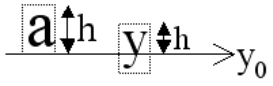

Figure 4. Symbol structural information

- A language model (LM): LM defines the grammar that produces acceptable mathematical expressions. Since two-dimensional grammars are faced to performance issues, we have described one as a set of one-dimensional rules on both vertical and horizontal axes. Vertical rules (VR) and horizontal rules (HR) are applied successively until elementary symbols are reached to perform a bottom up parsing algorithm.
- A decision maker (DM): DM organizes all the SHs, and selects the one that minimizes the cost function and respect the language model in order to represent a validate expression.

From a computing perspective, it can be considered as a Dynamic Programming (DP) algorithm, which is well adapted to this kind of decision making problems [16]. However, the key point is that this is not a standard 1D-DP where only consecutive strokes can be considered at a time, but it is an extension to a 2D-DP. To avoid the combinatory explosion of the dimension of the search space, some constraints are added. They limit the maximum number of strokes in one symbol, and also they limit the number of time jumps inside one symbol. Maximum number of hypothesis is limited also in addition to maximum distances between strokes forming the same hypothesis.

3.2. Global learning

We adopt a global learning method to train the classifier directly from mathematical expressions instead of using a pre-trained recognizer from a set of isolated symbols. We chose a multi-layer perceptron neural network (MLP) as a classifier. In this case, a gradient-based backpropagation algorithm is used. It takes into account the ground truth of the given ME (ideal segmentation and corresponding labels of symbols) and the best current interpretation resulting from a specific segmentation, and corresponding recognized symbols.

With the global learning, the classifier is trained to recognize symbols as they really appear in expressions instead of their isolated counterpart. Another advantage is that we can handle situations where the 2D segmentor (SHG) gives a hypothesis of stroke grouping that does not represent symbols; it is considered as a "Junk" situation. Though, global learning trains the classifier to recognize an additional special class, we call it: "Junk class". In addition, global learning considers symbols distribution in expressions assuring better recognition performance.

3.3. Expression recognition process

As shown in figure 5, expression recognition process is done as follows. The hypothesis generator produces a set of symbol hypothesis (SH). For each SH_i, a local cost function is computed, it is composed of two parts, recognition cost C_r and structural cost C_s that are linearly combined using a weighting coefficient λ, which has been set experimentally:

$$C(SH_i) = \lambda . C_r(SH_i) + C_s(SH_i) \qquad (1)$$

The recognition cost is measured on a negative log scale depending on the recognition score, which is available as the probability $p(SH_i)$ of a hypothesis being the recognized symbol. Hypothesis strokes number n is also used to weight the recognition score, which is:

$$C_r(SH_i) = -\log(p(SH_i)).n^{\alpha} \qquad (2)$$

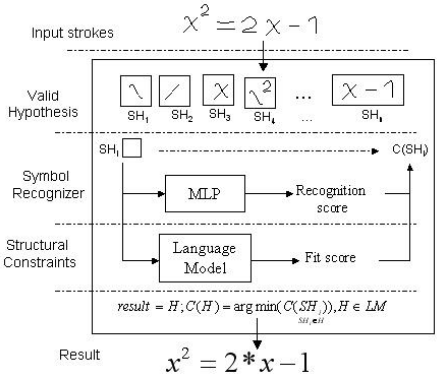

Figure 5. Mathematical expression recognition example

The parameter alpha allows tuning the favorable number of strokes per hypothesis. A large alpha value favors hypothesis with smaller number of strokes.

On the other hand, structural cost represents how good the hypothesis fit to both the layout of the whole expression and the context. Calculating this cost requires a structural analysis of the expression using h and y_0 values. A language model is also used to respect mathematical expression producing rules. Thus invalid expressions are not accepted. In the next section, we present some results of the preliminary tests.

4. Experiments

Expressions corpus is recovered from previous work of Raman [17], through the Aster mathematical expression base.

Aster base consists of 62 different expressions covering many domains with an average length of 13 symbols per ME. The total number of symbols is 839 symbols within 48 distinct classes including digits, Roman letters, Greek letters, binary operators, elastic symbols and functions.

With respect to the current grammar rules that we have implemented, we considered a sub-group of the Aster base consisting of 36 expressions covering all common math disciplines of the Aster base, table 1. For this subset, number of classes is restricted to 34 symbols.

Table 1 - Corpus extracted from Aster base

Domains	# Expressions
Simple fractions and expressions	8
Examples of Knuth	6
Continuous fractions	1
Algebraic expressions	3
Square roots	2
Trigonometric identities	6
Logarithms	3
Series	3
Integrals	1
Summations	2
Hyperbolic functions	1
Total	36

Since there is no public database of online mathematical expressions, we used a tool developed by our research group "Latex2Ink" [18]. This tool allows us to produce any corpus of handwritten mathematical expressions starting from previously collected isolated symbols. It generates pseudo-synthetic handwritten mathematical expressions using a stochastic layout guided by the Latex string defining the expression. In addition, we have also collected a set of real expressions.

4.1. Synthetic expression base

A previously isolated symbols base of 280 different writers was used to generate a synthetic ME train base, using 180 writers, and a test expression base, using the remaining 100 writers, referred as Synthetic expressions in table 4. Table 2 shows the constitution of both train and test expression bases. Figure 6 shows some examples of generated expressions.

Table 2. Synthetic expression base constitution

	# Writers	# Expressions	# Symbols
Train	180	$180 \times 36 = 6480$	$180 \times 412 = 74160$
Test	100	$100 \times 36 = 3600$	$100 \times 412 = 4120$

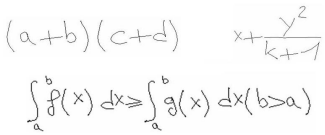

Figure 6. Examples of synthetic expressions

However, generated expressions are not intended to replace real ones; it aims to provide a large quantity of examples to be able to train and tune the system.

4.2. Collected expressions base

Additionally, the system needs to be tested with a real set of online handwritten mathematical expressions. Each of the 36 expressions has been written by two different writers forming a new database of 72 expressions to test the system. Ten writers have been involved in this second dataset, referred as Real expressions in table 4.

5. Results

The evaluation of the system at the ME level is too global to be significant. It is necessary to give performances at some intermediate levels to have a better insight of the actual behavior of the system. Thus, we chose three measurements similar to those used recently in [6, 19] in order to be able to compare our results, keeping in mind that expressions test bases are different. Experiments were carried out with a MLP architecture, using 100 neurons in the hidden layer and seven local features for every re-sampled points (30) along the trajectory of each hypothesis, resulting in a 210 dimension input vector.

5.1. Isolated symbol recognizer performance

Although the classifier will not be used later in the context of isolated symbols, we give in this section for comparison purpose, the performances that it achieved when trained with the isolated symbols coming from the 180 writer dataset. We obtain a recognition rate of 95.4% on the test set (34 classes). This result is compared with [19] in table 3.

Table 3. Isolated symbol recognition performance

Recognizer	Class number	Test %
[19]	99	87.5
MLP_100	34	95.4

5.2. Expression recognizer performance

The recognizer performance was evaluated in table 4 by these three different measurements:

- Segmentation rate (SegRate),
- Recognition rate (RecRate),
- Expression recognition rate (ExpRate),

representing respectively the percentage of correctly segmented symbols, correctly recognized symbols and expressions totally correctly interpreted.

Table 4. Expression recognition rates on test dataset

		SegRate %	RecoRate %	ExpRate %
Average rate in [19]		94.8	84.8	29.2
MLP trained on isolated symbols	Synthetic expressions	75	68.2	32.1
	Real expressions	49.7	46.8	12.9
MLP trained globally with synthetic expressions	Synthetic expressions	90.6	81.6	56
	Real expressions	91.2	75	37.1

Table 4 shows that there is a large improvement when comparing the results obtained with the classifier trained on the isolated symbols or trained in the global system. In this latter case, it can learn the incorrect segmentation hypothesizes using the "junk" class and adapt itself to the actual context of recognition within segmentation and interpretation. We are able to obtain competitive results due to the simultaneous optimization of segmentation, recognition and interpretation.

6. Conclusion and perspective

In this paper, we have presented a new framework of mathematical expression recognition. We propose a global learning method instead of using pre-trained classifiers. Furthermore, our system was not only trained and tested by a large number of artificially generated expressions but also tested by real complex expressions. In this case, the segmentation rate reaches 91.2%, the symbol recognition rate being 75%, and the global ME recognition rate 37.1%. Our experiments show promising results in the area of handwritten mathematical expressions compared with recent results.

7. References

[1] Dorothea Blostein, A.G., Recognition of mathematical notation, in *Handbook on Optical Character Recognition and Document Image Analysis*, Q.s.U. Department of Computing and Information Science, 1997, World Scientific Publishing Company: Kingston, Ontario, Canada. p. 557-582.

[2] Kam-Fai Chan, D.-Y.Y., An efficient syntactic approach to structural analysis of on-line handwritten mathematical expressions, *Pattern Recognition*, 2000. 33: p. 375 - 384.

[3] C. Faure, Z.X.W, Automatic perception of the structure of handwritten mathematical expressions, in *Computer Processing of Handwriting*, 1990, World scientific, Singapore.

[4] J. Ha, R.M.H., and I. T. Phillips, Understanding mathematical expressions from document images, ICDAR. 1995, p. 956-959.

[5] Steve Smithies, K.N., James Arvo. A Handwriting-Based Equation Editor, the Graphics Interface, 1999, Kingston, Ontario, Canada.

[6] Ryo Yamamoto, S.S., Takuya Nishimoto, Shigeki Sagayama. On-Line Recognition of Handwritten Mathematical Expressions Based on Stroke-Based Stochastic Context-Free Grammar, IWFHR 2006, La Baule, France.

[7] Nakayama, Y. A prototype pen-input mathematical formula editor, in *EDMEDIA*, 1993.

[8] A. Belaid, J.-P.H, A syntactic approach for handwritten mathematical formula recognition, in Transactions on Pattern Analysis and Machine Intelligence, 1984.

[9] Marzinkewitsch, R. Operating computer algebra systems by handprinted document, International Symposium on Symbolic and Algebraic Computation, 1991.

[10] Yannis A. Dimitriadis, J.L.C., Towards an ART based mathematical editor that uses online handwritten symbol recognition, *Pattern Recognition*, 1995. 28(6): p. 807 822.

[11] Stefan Lehmberg, H.-J.W., Manfred Lang. A Soft-decision approach for symbol segmentaiton within handwritten mathematical expressions, Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1996, Atlanta, USA.

[12] R. Fukuda, S.I., F. Tamari, M. Xie, and M. Suzuki. A technique of mathematical expression structure analysis for the handwriting input system, ICDAR, 1999.

[13] Martin, W. Computer input/output of mathematical expressions, in *Symbolic and Algebraic Manipulations*, 1971, New York.

[14] Daniel Prusa, V.H. 2D Context-Free Grammars: Mathematical Formulae Recognition. in The Prague Stringology Conference, 2006, Prague.

[15] Percy Liang, M.N., Michael Shilman, and Paul Viola. Efficient Geometric Algorithms for Parsing in Two Dimensions, ICDAR 2005, Seoul, South Korea.

[16] M. Held, R.M.K., The construction of discrete dynamic programming algorithms, IBM Syst. J. 4 (2), 1965: p. 136-147.

[17] Raman, T.V., Audio system for technical readings, 1994, Cornell Universtiy.

[18] Awal A.M., Cousseau R., Viard-Gaudin C. Convertisseur d'équations LATEX2Ink,. in Colloque International Francophone sur l'Ecrit et le Document 2008, Rouen, France.

[19] Taik Heon Rhee, K.-E.K., Jin Hyung Kim. Robust Recognition of Handwritten Mathematical Expressions Using Search-based Structure Analysis, ICFHR, 2008, Montreal.